DOCUMENT RESUME

ED 064 960 24 EM 010 210

TITLE	EIN Software Catalog; Entries, Volume III.
INSTITUTION	Interuniversity Communications Council (EDUCOM), Boston, Mass.
SPONS AGENCY	National Science Foundation, Washington, D.C.; Office of Education (DHEW), Washington, D.C.
BUREAU NO	BR-8-0697/Attachment No. 1
PUB DATE	[72]
GRANT	OEG-0-8-080697-4468(007)
NOTE	563p.; See also EM 010 206, EM 010 207, EM 010 208, EM 010 209, EM 010 211, EM 010 212
EDRS PRICE	MF-$0.65 HC-$19.74
DESCRIPTORS	Behavioral Sciences; Biological Sciences; *Catalogs; Computer Oriented Programs; *Computer Programs; Earth Science; Educational Administration; Engineering; *Information Networks; Information Science; *Interinstitutional Cooperation; Library Science; Mathematics; Operations Research; Physical Sciences; *Program Descriptions; Statistics; Technology; Tests
IDENTIFIERS	*Educational Information Network; EIN

ABSTRACT

The EIN (Educational Information Network) is a non-profit operation which coordinates the sharing of educational computing resources. It is administered by EDUCOM and funded jointly by the U. S. Office of Education and the National Science Foundation. EIN maintains a group of contact personnel at member institutions to serve as a liaison between the institution and EIN. Through these persons items of software are offered for distribution. EIN also publishes a catalog of the software which is available through the network. The four- volume catalog contains an alphabetical listing of the participating EIN members that are represented in the catalog by program descriptions; descriptions of each computer facility listed and its general pricing algorithm; abstracts of available programs, subdivided into 13 areas of application; three indexes--by EIN number, by descriptive title, and by keyword; and complete descriptions of programs, including user instructions, samples of input and output, and cost estimates. Volume one contains a description of the EIN, facilities descriptions, abstracts, and indexes. This volume is one of three which contain the complete program descriptions. (JY)

FILMED FROM BEST AVAILABLE COPY

ED 064960

EIN Software Catalog; Entries, Volume III.

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

EM010 210

DESCRIPTIVE TITLE	Static Leontief Input—Output Analysis
CALLING NAME	INOUT
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	M.C. Hallberg M. Swope Department of Agricultural Economics, The Pennsylvania State University
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Daniel Bernitt, 105 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

INOUT will solve the static Leontief input—output problem for a model with as many as 175 sectors. From data consisting of a square input matrix, an output vector, and a set of final demand vectors, a great variety of results may be selectively opted, calculated, and printed. Among these are matrices of technical and interdependency coefficients, matrices of interdependency values and net effects, and some other useful vectors.

REFERENCE

Leontief, W., *et al.*, *Studies in the Structure of the American Economy*, Harvard Economic Research Proj. (Oxford University Press, New York, 1953); theoretical and empirical explorations in input—output analysis.

USER INSTRUCTIONS

The data deck consists of the following cards in the order specified.

```
--------JOB CARD--------
// EXEC LIBRARY.PARM=INOUT
// DATA.INPUT DD *
Master control card for Problem 1
Total output or receipts vector
Input matrix cards
Final demand vector
Master control card for Problem 2
              .
              .
              .
              .
```

Master Control Card

Columns	*Contents*
1– 3	number of sectors (must not exceed 175)
10	1 if technical coefficients are to be printed (else =0)
15	1 if interdependency coefficients are to be printed (else = 0)
16–20	number of final demand vectors to be read
25	1 if sector multipliers are to be printed (else = 0)
30	1 if matrix of interdependency values is to be printed (else = 0)
35	1 if matrix of net effects is to be printed (else = 0)
40	1 if column sums of technical coefficients are to be printed (else = 0)
45	1 if interdependency coefficients are to be punched (else = 0)
50	1 if matrix of technical coefficients is to be punched in 8F10.5 format (else = 0)
55	1 if matrix of interdependency coefficients is to be punched in 4F20.8 format (else = 0)

Total Output or Receipts Vector Cards

This vector is punched eight elements per card in 10-column fields. If the decimal point is not punched, it will automatically be

continued

placed between the sixth and seventh columns in each 10-column field. These cards are to follow the master control card.

Input-Matrix Cards

The input matrix is punched by rows, eight elements per card in 10-column fields. The first element in each row will occupy the first 10-column field on a new card—each subsequent element in that row will follow in consecutive order. If the decimal point is not punched, it will automatically be placed between the sixth and seventh columns in each 10-column field. These cards are to follow the total output vector cards.

Final Demand Vector(s) Cards

If a nonzero number appears in Cols. 16–20 of the master control card, the final demand vectors desired will be punched in the same fashion as the total output or receipts vector. Each demand vector will require a separate set of cards. These cards will follow the input-matrix cards. To obtain solutions for more than one final demand vector, insert a set of cards for each final demand vector and make the appropriate entry in Cols. 16–20 of the master control card.

SAMPLE INPUT

```
//              'S4002.T=200.K=2000'.'BERNILLI'
// EXEC LIBRARY.PARM=INPUT
//DATA.INPUT DD *
   24     1     1    1     1    1    1     1    1     1    1
1790.      852.       999.       2274.      123.       137.       355.       805.
778.       1692.      1076.      1121.      751.       351.       204.       140.
95.        1137.      219.       192.       824.       166.       196.       452.
104.       5545.      311.       1122.      1612.
58.        312.                  17.
           1.

           2.

                      30.                   2.         2.
                                            10.
           26.        1.         5.         8.



10.                              107.
16.
1.                                                                1.
           2.
           21.        4.
           1.         5.         5.         5.         2.
           6.
           29.        2.         6.         8.
           11.                   7.                    5.

                                                                             4.
           24.        1.         6.         7.
60.                   1.         16.                                         1.
                      6.         2.         7.         5.                    1.
                 .                  .                  .
                 .                  .                  .
                 .                  .                  .
15.        376.       618.       495.       32.        10.        81.        45.
166.       101.       52.        55.        61.        29.        18.        10.
17.        57.        53.        72.        435.       48.        79.        368.
98.
14.                   16.                   13.                              141.
1.         8.         4.         3.         1.                    2.
           9.         10.        1.         2.         1.                    1.
6.         40.        2.         8.         12.


                                                       11.                   62.





177.       25.                   559.       3.         17.        8.         28.
165.       97.        25.        27.        96.        38.        76.        79.
7.         185.

1400.      768.       999.       2142.      29.        72.        156.       111.
233.       788.       321.       326.       385.       345.       71.        10.
86.        435.       71.        119.       613.       52.        13.        254.
64.        2154.      16.        1069.
```

SAMPLE OUTPUT

```
        29    SECTOR INPUT-OUTPUT MODEL

                                A=MATRIX OF TECHNICAL COEFFICIENTS
                                (I-A) INVERSE=MATRIX OF INTERDEPENDENCE COEFFICIENTS



     TECHNICAL COEFFICIENTS


SECTOR    1    0.03240223    0.36619718    0.0           0.00745941    0.0           0.0           0.0
               0.0           0.0           0.00059102    0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.00036069    0.0           0.0
               0.0
SECTOR    2    0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.02788104    0.0           0.00266312    0.00569801
               0.0           0.0           0.0           0.0           0.0           0.0           0.01206273
               0.0           0.0           0.0           0.0           0.00468891    0.00321543    0.00445633
               0.00496278
SECTOR    3    0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0
SECTOR    4    0.00558659    0.0           0.0           0.04695042    0.0           0.0           0.0
               0.0           0.02056555    0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.01052632    0.0           0.0           0.0           0.0
               0.0           0.00510204    0.0           0.0           0.00036069    0.0           0.0
               0.0

                                 .                           .             .
                                 .                           .             .
                                 .                           .             .

SECTOR   26    0.00837989    0.44131455    0.61861862    0.21720053    0.26016260    0.07299270    0.22816901
               0.05593062    0.21336761    0.05969267    0.04832714    0.04906334    0.08122503    0.08262108
               0.08866995    0.07142857    0.17894737    0.05013193    0.24200913    0.37500000    0.52472859
               0.28915663    0.40306122    0.81415929    0.94230769    0.0           0.0           0.0
               0.0
SECTOR   27    0.00782123    0.0           0.01601602    0.0           0.10569106    0.0           0.0
               0.17515528    0.00128535    0.00472813    0.00371747    0.00267618    0.00133156    0.0
               0.00985222    0.0           0.0           0.00791557    0.04566210    0.00520833    0.00241255
               0.00602410    0.0           0.00221239    0.05769231    0.00721371    0.00643087    0.00713012
               0.00744417
SECTOR   28    0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.06626506    0.0           0.13716814    0.0           0.0           0.0           0.0
               0.0
SECTOR   29    0.09888268    0.02934272    0.0           0.24528302    0.02439024    0.12408759    0.02253521
               0.03479261    0.21209226    0.05732861    0.02323420    0.02408564    0.12782956    0.10826211
               0.37438424    0.56428571    0.07368421    0.16270888    0.0           0.0           0.0
               0.0           0.0           0.0           0.0           0.0           0.0           0.0
               0.0
```

continued

```
        COLUMN SUMS


SECTOR    1          0.48044693
SECTOR    2          0.90375587
SECTOR    3          0.76176176
SECTOR    4          0.67485739
SECTOR    5          0.58536585
SECTOR    6          0.50364964
SECTOR    7          0.26450704
SECTOR    8          0.27701863
SECTOR    9          0.55526892
SECTOR   10          0.15307329
SECTOR   11          0.17657953
SECTOR   12          0.34875572




SECTOR   13          0.53335473
SECTOR   14          0.37606833
SECTOR   15          0.64035409
SECTOR   16          0.72857143
SECTOR   17          0.68421053
SECTOR   18          0.29139648
SECTOR   19          0.55251142
SECTOR   20          0.61979167
SECTOR   21          0.66586248
SECTOR   22          0.61445783
SECTOR   23          0.47959184
SECTOR   24          0.97345133
SECTOR   25          1.00000000
SECTOR   26          0.46438233
SECTOR   27          0.47905568
SECTOR   28          0.46078431
SECTOR   29          0.46464320
```

continued

```
      INTERDEPENDENCE COEFFICIENTS


SECTOR   1      1.03438373     0.38063829     0.00256198     0.01011666     0.00155642     0.00096610     0.00098055
                0.00083736     0.00192827     0.00108026     0.01096968     0.00057936     0.00239391     0.00333841
                0.00184831     0.00230108     0.00172055     0.00085061     0.00146243     0.00180913     0.00689721
                0.00231460     0.00171884     0.00348265     0.00364993     0.00370152     0.00280732     0.00322592
                0.00343522
SECTOR   2      0.00190929     1.00502373     0.00593744     0.00494629     0.00368833     0.00215870     0.00230411
                0.00209732     0.00404009     0.00114223     0.02881572     0.00102574     0.00581613     0.00858199
                0.00462414     0.00582894     0.00391589     0.00210231     0.00349484     0.00416493     0.01751222
                0.00560123     0.00390490     0.00820051     0.00851602     0.00859609     0.00720826     0.00833566
                0.00888482
SECTOR   3      0.0            0.0            1.00000000     0.0            0.0            0.0            0.0
                0.0            0.0            0.0            0.0            0.0            0.0            0.0
                0.0            0.0            0.0            0.0            0.0            0.0            0.0
                0.0            0.0            0.0            0.0            0.0            0.0            0.0
                0.0
SECTOR   4      0.00653338     0.00322249     0.00317652     1.05036082     0.00050291     0.00025657     0.00031860
                0.00033586     0.02226208     0.00015749     0.00019625     0.00019527     0.00066165     0.00115349
                0.00166979     0.00104729     0.01331421     0.00028879     0.00053072     0.00467103     0.00262050
                0.00125454     0.00610881     0.00114669     0.00118505     0.00120521     0.00085580     0.00080132
                0.00082736
                              .                           .              .
                              .                           .              .
                              .                           .              .
SECTOR  26      0.06990238     0.52883257     0.70792589     0.32367989     0.32475914     0.12159109     0.25738333
                0.08457522     0.27036824     0.07602065     0.07898364     0.08029718     0.19399118     0.15516562
                0.15671866     0.13890592     0.29357831     0.07969418     0.30429585     0.46677207     0.60635056
                0.43589040     0.46643836     0.85812006     1.02885815     1.08566967     0.10093665     0.08181874
                0.08613992
SECTOR  27      0.01803351     0.02340881     0.03568461     0.01615946     0.12145349     0.00555856     0.00790292
                0.18317116     0.01500019     0.00914030     0.00823690     0.00874772     0.01499922     0.01030009
                0.02793395     0.01980566     0.01419491     0.01765220     0.05832627     0.01943185     0.01890793
                0.02059282     0.01225883     0.02830878     0.08489824     0.02717748     1.02767074     0.02693400
                0.02743623
SECTOR  28      0.00123875     0.00300773     0.00257525     0.00267795     0.00326833     0.00199000     0.00151194
                0.00103145     0.00203759     0.00075724     0.00049592     0.00068338     0.01126754     0.00251738
                0.00259602     0.00237497     0.00362450     0.00105664     0.00133712     0.00215195     0.00212765
                0.06805823     0.00291856     0.14055858     0.00356711     0.00354560     0.00391837     1.00338339
                0.00356517
SECTOR  29      0.14259887     0.10955007     0.05783848     0.30805483     0.05522211     0.15369011     0.03883072
                0.04966159     0.25067870     0.96627746     0.03629419     0.04657904     0.16168981     0.13774265
                0.42344573     0.60582200     0.15387606     0.18001291     0.04669637     0.07315905     0.05036813
                0.04240869     0.02546786     0.04740762     0.04702945     0.04696867     0.04802220     0.04675191
                1.04674054
```

continued

```
     SECTOR MULTIPLIERS


SECTOR    1             1.75845171
SECTOR    2             2.61751947
SECTOR    3             2.37829803
SECTOR    4             2.24202846
SECTOR    5             2.01196011
SECTOR    6             1.80469549
SECTOR    7             1.50609016
SECTOR    8             1.50195679
SECTOR    9             1.97595691
SECTOR   10             1.26972594
SECTOR   11             1.31425545
SECTOR   12             1.49337767
SECTOR   13             1.92883132
SECTOR   14             1.68236403
SECTOR   15             2.13144021
SECTOR   16             2.27832987
SECTOR   17             2.20212481
SECTOR   18             1.50498571
SECTOR   19             1.92131127
SECTOR   20             2.14093116
SECTOR   21             2.19941974
SECTOR   22             2.16939559
SECTOR   23             1.87105213
SECTOR   24             2.72894405
SECTOR   25             2.78221157
SECTOR   26             1.77989373
SECTOR   27             1.82006961
SECTOR   28             1.76923921
SECTOR   29             1.78097197
```

continued

```
     OUTPUT LEVELS REQUIRED TO MEET SPECIFIED FINAL DEMAND



                        FINAL DEMAND              REQUIRED OUTPUT


SECTOR    1            1400.00000000              1790.00000000
SECTOR    2             768.00000000               852.00000000
SECTOR    3             999.00000000               999.00000000
SECTOR    4            2142.00000000              2279.00000000
SECTOR    5              29.00000000               123.00000000
SECTOR    6              72.00000000               137.00000000
SECTOR    7             156.00000000               355.00000000
SECTOR    8             111.00000000               805.00000000
SECTOR    9             233.00000000               778.00000000
SECTOR   10             788.00000000              1692.00000000
SECTOR   11             321.00000000              1076.00000000
SECTOR   12             326.00000000              1121.00000000
SECTOR   13             385.00000000               751.00000000
SECTOR   14             345.00000000               351.00000000
SECTOR   15              71.00000000               203.00000000
SECTOR   16              10.00000000               140.00000000
SECTOR   17              86.00000000                95.00000000
SECTOR   18             435.00000000              1137.00000000
SECTOR   19              71.00000000               219.00000000
SECTOR   20             119.00000000               192.00000000
SECTOR   21             613.00000000               829.00000000
SECTOR   22              52.00000000               166.00000000
SECTOR   23              13.00000000               196.00000000
SECTOR   24             254.00000000               452.00000000
SECTOR   25              64.00000000               104.00000000
SECTOR   26            2154.00000000              5545.00000000
SECTOR   27              16.00000000               311.00000000
SECTOR   28            1049.00000000              1122.00000000
SECTOR   29               0.0                     1612.00000000

TOTAL                 13082.00000000             25432.00000000
```

continued

```
MATRIX OF INTERDEPENDENCY VALUES


1448.13722798      292.33020774       2.55942063        21.66988022        0.04513629         0.06955907
   0.15296646        0.09294674       0.42598658         0.85124246        3.52094700         0.18896975
   0.92165539        1.15175105       0.13123013         0.02301079        0.14796768         0.37001354
   0.10383272        0.21528658       4.22799155         0.12035930        0.02234496         0.88459254
   0.23359558        7.97307300       0.04491710         3.38398716        0.0
1790.00000000
   2.67301224      771.85522805       5.93150415        10.59494334        0.10696143         0.15542670
   0.35944161        0.22280305       0.94134157         0.90007714        9.24984614         0.33439202
   2.23920941        2.96078739       0.32831377         0.05828943        0.33676693         0.91450623
   0.24813388        0.49562697      10.73499386         0.29126399        0.05076368         2.08292847
   0.54502527       18.51597119       0.11533211         8.74410998        0.0
 852.00000000
   0.0               0.0            999.00000000         0.0               0.0                0.0
   0.0               0.0              0.0                0.0               0.0                0.0
   0.0               0.0              0.0                0.0               0.0                0.0
   0.0               0.0              0.0                0.0               0.0                0.0
   0.0               0.0              0.0                0.0               0.0
 999.00000000
   9.70673741        2.47487068       3.17334603      2249.97286799        0.01458451         0.01847320
   0.04970153        0.03729079       5.18706435         0.12410067        0.06299490         0.06365908
   0.25473516        0.39795271       0.11855513         0.01047294        1.14502169         0.12562253
   0.03768103        0.55585200       1.60636589         0.06523597        0.07941447         0.29125941
   0.07584339        2.59602511       0.01369282         0.84058858        0.0

                         .                .                .
                         .                .                .
                         .                .                .

  97.86333820      406.14341685     707.21796273       693.32232107        9.41801513         8.75455842
  40.15179984        9.38784968      62.99579946        59.90426875       25.35374751        26.17687947
  74.68660409       53.53213953      11.26902456         1.38905919       25.24773469        34.66696995
  21.60500505       55.54587652     371.69289081        22.66630072        6.06369866       228.12249473
  65.84692173     2338.53247509       1.61498634        85.82786122        0.0
5545.00000000
  25.24691192       17.97796320      35.64892335        34.61356654        3.52215123         0.68821608
   1.23285539       20.33199914       3.49504474         7.20255419        2.64404493         2.85175724
   5.77470140        3.55353175       1.98331080         0.19805656        1.22076200         7.67870776
   4.14116493        2.31239028      11.59049766         1.07082673        0.15936476         7.19042937
   5.43348737       58.54028138      16.44273177        28.25376754        0.0
 311.00000000
   1.73425192        2.30993838       2.57267498         5.73617676        0.09478167         0.14328011
   0.23586189        0.11449050       0.47475918         0.59670257        0.15919130         0.22278306
   4.34570168        0.86849764       0.18431769         0.02374967        0.31170717         0.45963727
   0.09493541        0.25607026       1.30424916         3.53902815        0.03794128        35.70187861
   0.22829488        7.63722585       0.06269394      1052.54917900        0.0
1122.00000000
 195.62442329       84.13445488      57.78063767       659.85343817        1.60144125        11.06568793
   6.05759169        5.51243611      58.40813631        52.22663690       11.65043367        15.18476676
  62.25057587       47.52121515      30.06464674         6.05822001       13.23334104        78.30561557
   3.31544192        8.70592654      30.87566074         2.20525192        0.33108222        12.04153541
   3.00988482      101.17051496       0.76835516        49.04264730        0.0
1612.00000000
```

continued

11

```
 MATRIX OF NET EFFECTS--(I-A) INVERSE MINUS A


   1.00158150      0.01444111      0.00256198      0.00265725      0.00155642      0.00096610
   0.00098055      0.00093736      0.00182827      0.00048924      0.01096868      0.00057936
   0.00239391      0.00333841      0.00184931      0.00230108      0.00172055      0.00085061
   0.00146243      0.00180913      0.00683721      0.00231460      0.00171884      0.00348265
   0.00364993      0.00334083      0.00280732      0.00322592      0.00343522
   0.00190929      1.00502373      0.00593744      0.00494629      0.00368833      0.00215870
   0.00230411      0.00209732      0.00404009      0.00114223      0.00093468      0.00102574
   0.00315301      0.00288399      0.00462414      0.00582894      0.00391589      0.00210231
   0.00349494      0.00416493      0.00544950      0.00560123      0.00390490      0.00820051
   0.00951602      0.00393718      0.00399282      0.00387933      0.00392204
   0.0             0.0             1.00000000      0.0             0.0             0.0
   0.0             0.0             0.0             0.0             0.0             0.0
   0.0             0.0             0.0             0.0             0.0             0.0
   0.0             0.0             0.0             0.0             0.0             0.0
   0.0             0.0             0.0             0.0             0.0
   0.00134679      0.00322249      0.00317652      1.00341040      0.00050291      0.00025657
   0.00031860      0.00033586      0.00169653      0.00015749      0.00019625      0.00019527
   0.00066165      0.00115349      0.00166979      0.00104729      0.00278789      0.00028879
   0.00053072      0.00467103      0.00262050      0.00125454      0.00100676      0.00114669
   0.00118505      0.00084453      0.00085580      0.00080132      0.00082736
                            .             .        .
                            .             .        .
                            .             .        .
   0.06152250      0.08751802      0.08930727      0.10647936      0.06459654      0.04859839
   0.02921432      0.02867460      0.05700063      0.01632797      0.03065650      0.03123384
   0.11276615      0.07254454      0.07004870      0.06747735      0.11463094      0.02956226
   0.06228671      0.09177207      0.08162197      0.14673377      0.06337713      0.08396077
   0.08655046      1.08566967      0.10093665      0.08181874      0.08613992
   0.01021228      0.02340881      0.01966859      0.01615946      0.01576243      0.00955856
   0.00790292      0.00801588      0.01371484      0.00441216      0.00451943      0.00607154
   0.01366767      0.01030009      0.01808174      0.01980566      0.01419491      0.00973663
   0.01266417      0.01422352      0.01649528      0.01456873      0.01225883      0.02609639
   0.02720593      0.01996377      1.02123987      0.01980388      0.01999206
   0.00123875      0.00300773      0.00257525      0.00267795      0.00326833      0.00199000
   0.00151194      0.00103145      0.00203759      0.00075724      0.00049592      0.00068338
   0.01128754      0.00251738      0.00259602      0.00237497      0.00362450      0.00105664
   0.00133712      0.00215185      0.00212765      0.00179317      0.00291856      0.00339044
   0.00356711      0.00354560      0.00391837      1.00338339      0.00356517
   0.04370619      0.08020735      0.05783848      0.06277181      0.03083187      0.02960252
   0.01629550      0.01487898      0.03859643      0.00894885      0.01305999      0.02249340
   0.03386025      0.02948054      0.04906149      0.04153629      0.08019185      0.01730403
   0.04669637      0.07315905      0.05036810      0.04240869      0.02546786      0.04740762
   0.04702945      0.04696867      0.04802220      0.04675181      1.04674094

***END OF RUN***
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 11 seconds. At the current rate for computer time ($0.11/ sec.) at The Pennsylvania State University, the chargeable time amounts to $1.21.

Charge to user = computer time + network overhead
= $1.21 + network overhead

CONTENTS—INOUT

pages	
1	Identification & Abstract
3– 4	User Instructions
5–12	I/O
13	Cost—Contents

DESCRIPTIVE TITLE	Synagraphic Mapping
CALLING NAME	SYMAP
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Laboratory for Computer Graphics Harvard University Graduate School of Design Adapted by Larry Rich Jeffrey Simon Larry Sinkey Department of Architectural Engineering, The Pennsylvania State University
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks, listings, and documentation presently available
CONTACT	Dr. Daniel Bernitt, EIN Technical Representative, 105 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

The Synagraphic Mapping program (SYMAP) produces maps that depict spatially disposed quantitative and qualitative information. Raw data of every kind (physical, social, economic, etc.) may be related, weighted, and aggregated in a graphic format by assigning values to the coordinate locations of data points or data zones. According to the application and desired representation of data, three basic types of mapping procedure may be specified: contour, conformant, or proximal.

CONTOUR
—based on the use of *contour* lines, each of which represents a value remaining constant throughout its length. The map consists of closed curves that connect all points having the same numerical value. The value at each of the different levels of contour

continued

(where a single contour level will represent a particular data value) is determined by the program, according to the scale of the map and the range of the data. Between any two contour lines, a continuous variation is assumed. Therefore, the use of contour mapping should be restricted to the representation of continuously varying information, such as topography, rainfall, or population density.

CONFORMANT
—based on the *conformance* to the boundaries of a data zone. This type of mapping is best suited to data for which the representation as a continuously varying surface is inappropriate owing to the significance of areal limits or boundaries. Each predefined data zone is assigned one data value and, depending on its numeric class (range), one representative character on the map itself. Local variation within the zone boundaries will not be apparent, but will, on the average, be correct.

PROXIMAL
—based on *proximity* to a data point. In appearance, this type of map is similar to the conformant map. However, point information is used here to define the data zones. Each character location on the output map is assigned the value of the *nearest* data point, using nearest-neighbor techniques. Boundaries are then assumed along the lines where these values change. Then, the mapping is carried out as in the conformant type.

While the contour type of map is mose often used and the easiest to produce, the conformant and proximal maps are often more helpful in the "soft" disciplines. Output is in the form of printed pages that, if the total map size exceeds the width of the computer printed page (13 in.), may easily be glued or pasted together to form a continuous map. Also included in the output is a histogram showing frequencies for given data levels, plus several optional features.

REFERENCES

Robertson, J.C., "The SYMAP Programme for Computer Mapping," Cartographic 108–113 (Dec. 1967); taken from a report of the Select Committee on the Ordinance Survey of Scotland.

Fisher, M., "The Laboratory for Computer Graphics," Harvard Univ. Grad. School Design Suppl. (summer 1967).

continued

Shepard, D., "A Two-Dimensional Interpolation Function for Irregularly Spaced Data," Harvard Univ. Grad. School Design Lab. Computer Graphics (Feb. 1968); available from the EIN Office for the cost of duplication and mailing. Deals with the subject of analyzing irregularly spaced data derived from a continuous surface. A method is developed for reconstructing the surface from the sampled data. This method is the main device used to generate the maps produced by SYMAP.

USER INSTRUCTIONS

The user must establish the geographic area that he wishes to map, called the *study area*; it may be of any desired shape. He must then establish the relative locations of any controlling points to which his data relate. For these two purposes, it is best to construct a *source map*, with borders from which measurements may be taken. This source map may be identical in size to the maps that are to be produced—or it may be of any size, because there are automatic scaling and proportioning electives built into the program.

Owing to the availability of the digitizer at The Pennsylvania State University (PSU), it is recommended that users of the PSU modification of SYMAP consider using this device for these proportional adjustments. The purpose of the digitizer is to transfer automatically to punched cards, the X and Y coordinates from a map or drawing. The drawing or map itself is limited to 30X30 in., but it may be drawn to almost any scale. The digitizer is in *no way* connected to a computer but only provides the cards (via an IBM 029 card-punch machine) that can be used as computer input. Interested persons are directed to contact the EIN technical representative at PSU.

Input to the computer in the form of a deck of punched cards must be prepared. This deck will consist of certain introductory cards and a number of packages, each composed of additional cards covering a specific category of information about the map to be produced.

Job Card

Will be prepared by Computation Center personnel.

Introductory Cards

See Ref. 1.

Types of Packages

The titles of all available packages, with a brief explanation of their general purpose, are listed below in the sequency of their position in the deck. For more-complete and -definitive instructions in the preparation of these packages, see Ref. 1.

continued

A-OUTLINE

—describes the outline of the study area if nonrectangular by specifying the coordinate locations of the outline vertices. Used for contour and proximal maps only.

A-CONFORMOLINES

—used to give the positions of the data zones to which your data are to be related, by specifying the coordinate locations of vertices on the zonal outlines. This package is required for a conformant map.

B-DATA POINTS

—used to give the positions of the data points to which your data are to be related, by specifying their coordinate locations. Data points may be either the points for which data are available or the centers of areas, called *data zones*, for which data are available. (When warranted by the nature of your study, and under exceptional circumstances, other "centers" may be used, such as centers of population.) This package is required for contour and proximal maps.

C-OTOLEGENDS

—used to specify the relative position of legends that are to be adjusted automaticlly if the size and/or scale of the map are altered.

D-BARRIERS

—used to give the coordinate location and strength of impediments to interpolation at specified vertices.

E-VALUES

—used to assign numerical data to the data points and/or data zones by specifying the values involved. All such data must, of course, be measured on a consistent uniform basis. While normally required, this package may be omitted if you wish to procure a preliminary base map for checking locations before applying values.

E1-VALUES INDEX

—used to adjust the reference order of data values in the E-VALUES package.

F-MAP

—used to specify below the map an appropriate title for the identification of each separate map that you wish to run. In addition, it instructs the computer to make each specific map pursuant to

continued

certain electives. These electives provide a variety of options for obtaining maps suited to your particular needs. An F-MAP package is required for each map desired.

REFERENCES

1. *Introductory Manual for Synagraphic Computer Mapping*—"SYMAP," Pa. State Univ. Dept. Arch. Eng. & Grad. Interdisciplinary Program in Regional Planning. Rept. No. 69–4, Version 5, PSU Modification 3. Available from the EIN office for the cost of duplication and mailing.

SAMPLE INPUT—Population-Density Contour Map of Pennsylvania

```
               'C5993,T=200,R=600,SYMAP','HULL E J',MSGLEVEL=1
// EXEC LIBRARY,PARM=SYMAP
//DATA.INPUT DD *
VERSION 5
A-OUTLINES      0008      0052                                       01
                0027      0051                                       02
                0027      0059                                       03
                0028      0095                                       04
                0028      0131                                       05
                0028      0161                                       06
                0028      0197                                       07
                0028      0237                                       08
                0028      0278                                       50
                0031      0281                                       51
                  :         :                                         :
                0082      0304                                       68
                0087      0301                                       69

99999
B-DATA
                0018      0036                                       01
                0039      0082                                       02
                0042      0119                                       03
                0044      0140                                       04
                0046      0177                                       05
                  :         :                                         :
                0086      0085                                       24
                0096      0119                                       25
99999
C-OTOLEGENDS
   12     P     0165      0297
PHILADELPHIA
   10     P     0135      0034       -5.        -4.
PITTSBURGH
   15     P     0111      0149                                       10
*   PENN STATE
        TL      0099      0008                                       01
                0126      0046                                       02
                0138      0053                                       03
                0147      0054                                       04
                0157      0069                                       05
                  :         :                                         :
                0152      0295                                       19
                0155      0316                                       20
        IL      0082      0009                                       01
                0081      0043                                       02
                0084      0061                                       03
                0082      0068                                       04
                0084      0083                                       05
                  :         :                                         :
                0096      0283                                       21
                0096      0293                                       22
        SL      0087      0141                                       01
                0078      0148                                       02
                0074      0159                                       03
                0088      0167                                       04
                0081      0186                                       05
                  :         :                                         :
                0176      0227                                       29
                0190      0238                                       30
99999
```

continued

```
E-VALUES
               309.0
                50.0
                55.0
                15.0
                32.0
                48.0
                40.0
                38.0
                17.0
               517.0
                42.0
                13.0
                90.0
                42.0
                19.0
                46.0
                11.0
               117.0
                77.0
               187.0
               308.0
               144.0
                63.0
                72.0
                71.0
99999
F-MAP
C   PENNSYLVANIA POPULATION DENSITY
C   PERSONS PER SQUARE MILE

    1            24.       36.
    3            10.
    6            25.       25.       50.      100.      100.      100.
                100.      100.      400.    14584.
   10
1965 DATA
C. F. GERLACH
9999
   13            .11
   14            1.5       1.5       2.0       1.5
99999
/*
```

SAMPLE INPUT—Population-Density Conformant Map of Pennsylvania

```
//             'C5993,T=200,R=6000,SYMAP','HULL E J',MSGLEVEL=1
// EXEC LIBRARY,PARM=SYMAP
//DATA.INPUT DD *
VARSION 5
A-CONFORMOLINES
         A      0008      0052                                      01
                0027      0051                                      02
                0027      0059                                      03
                0037      0059                                      04
                0036      0012                                      05
                0027      0012                                      06
                0023      0023                                      07
                0008      0052                                      08
         A      0027      0059                                      01
                0028      0095                                      02
                0040      0095                                      03
                0040      0093                                      04
                0054      0093                                      05
99999
B-DATA
                0018      0036                                      01
                0039      0082                                      02
                  :         :                                        :
                0069      0219                                      12
                0081      0192                                      13
99999
C-OTOLEGENDS
   12    P      0165      0297
PHILADELPHIA
-  10    P      0135      0034         -5.      -4.
PITTSBURGH
   15    P      0111      0149                                      10
*   PENN STATE
        TL      0099      0008                                      01

                0143      0199                                      21
                0147      0198                                      22
                  :         :                                        :
                0176      0227                                      29
                0190      0238                                      30
```

continued

```
E-VALUES
               309.0
                50.0
                55.0
                15.0
                32.0
                48.0
                40.0
                38.0
                17.0
               517.0
                42.0
                13.0
                90.0
                42.0
                19.0
                46.0
                11.0
               117.0
                77.0
               187.0
               308.0
               144.0
                63.0
                72.0
                71.0
99999
F-MAP
C   PENNSYLVANIA POPULATION DENSITY
C   PERSONS PER SQUARE MILE

    1           24.       36.
    3           10.
    4           25.
    5         3000.
    6           25.       25.       50.      100.      100.      100.
               100.      100.      400.     2000.
   10
1965 DATA
C. F. GERLACH
9999
   13           .11
   14           1.5       1.5       2.0       1.5
   15            6.       10.
99999
/*
```

SAMPLE OUTPUT—Showing Contours of Population Density

```
A-OUTLINE
---------


VERTEX            DOWN     ACROSS

                  ISLAND    1

(    1)           27.00     51.00
(    2)            8.00     52.00
(    3)           27.00     59.00
(    4)           28.00     95.00
(    5)           28.00    131.00
(    6)           28.00    161.00
(    7)           28.00    197.00
(    8)           28.00    237.00
(    9)           28.00    278.00
              .
              .
              .

(   50)          177.00    268.00
(   51)          179.00    263.00
(   52)          182.00    260.00
(   53)          185.00    259.00
(   54)          186.00    194.00
(   55)          185.00     98.00
(   56)          183.00      7.00
(   57)           27.00     12.00
(   58)           23.00     23.00
(   59)            8.00     52.00
AREA=   45299.00           CENTER=(     105.65,     151.50)



B-DATA POINTS
-------------


POINT             DOWN     ACROSS

(    1)           18.00     36.00
(    2)           39.00     82.00
(    3)           42.00    119.00
(    4)           44.00    140.00
(    5)           46.00    177.00
(    6)           45.00    222.00
(    7)           40.00    251.00
(    8)           57.00    283.00
(    9)           73.00    306.00
(   10)           70.00    263.00
              .
              .
              .
(   60)          164.00    231.00
(   61)          143.00    250.00
(   62)          124.00    273.00
(   63)          144.00    292.00
(   64)          157.00    281.00
(   65)          168.00    268.00
(   66)          171.00    279.00
(   67)          169.00    291.00
```

continued

```
C-OTOLEGENDS
------------


VERTEX           DOWN    ACROSS     +ROWS      +COLS

(    1) 'PHILADELPHIA' ACROSS FROM

               165.00    297.00        0.         0.

(    2) 'PITTSBURGH' DOWN FROM

               135.00     34.00       -5.        -4.

(    3) '*   PENN STATE ' ACROSS FROM

               111.00    149.00        0.         0.

(    4) 'T' ON LINE


(    1)         99.00      8.00
(    2)        126.00     46.00
(    3)        138.00     53.00
(    4)        147.00     54.00
(    5)        157.00     69.00
           .
           .
           .
(   16)        153.00    194.00
(   17)        148.00    238.00
(   18)        161.00    267.00
(   19)        152.00    295.00
(   20)        155.00    316.00
LENGTH=    351.82

(    5) '1' ON LINE


(    1)         82.00      9.00
(    2)         81.00     43.00
(    3)         84.00     61.00
(    4)         82.00     68.00
(    5)         84.00     83.00
           .
           .
           .
(   16.         98.00    209.00
(   17)         94.00    228.00
(   18)         97.00    233.00
(   19)         93.00    265.00
(   20,         92.00    280.00
(   21)         96.00    283.00
(   22)         96.00    293.00
LENGTH=    297.99

(    6) 'S' ON LINE


(    1)         87.00    141.00
(    2)         78.00    148.00
(    3)         74.00    159.00
(    4)         88.00    167.00
(    5)         81.00    186.00
           .
           .
           .

           .
```

continued

```
F-VALUES
--------



DATUM          VALUE


(    1)         309.00
(    2)          50.00
(    3)          55.00
(    4)          15.00
(    5)          32.00
(    6)          48.00
(    7)          40.00
(    8)          38.00
(    9)          17.00
(   10)         517.00
             .
             .
             .
(   25)         161.00     212.00
(   26)         163.00     218.00
(   27)         169.00     223.00
(   28)         172.00     227.00
(   29)         176.00     227.00
(   30)         190.00     239.00
LENGTH=      219.26

             .
             .
             .
(   60)         294.00
(   61)         320.00
(   62)         654.00
(   63)         503.00
(   64)        1042.00
(   65)         277.00
(   66)        3005.00
(   67)       15584.00
```

continued

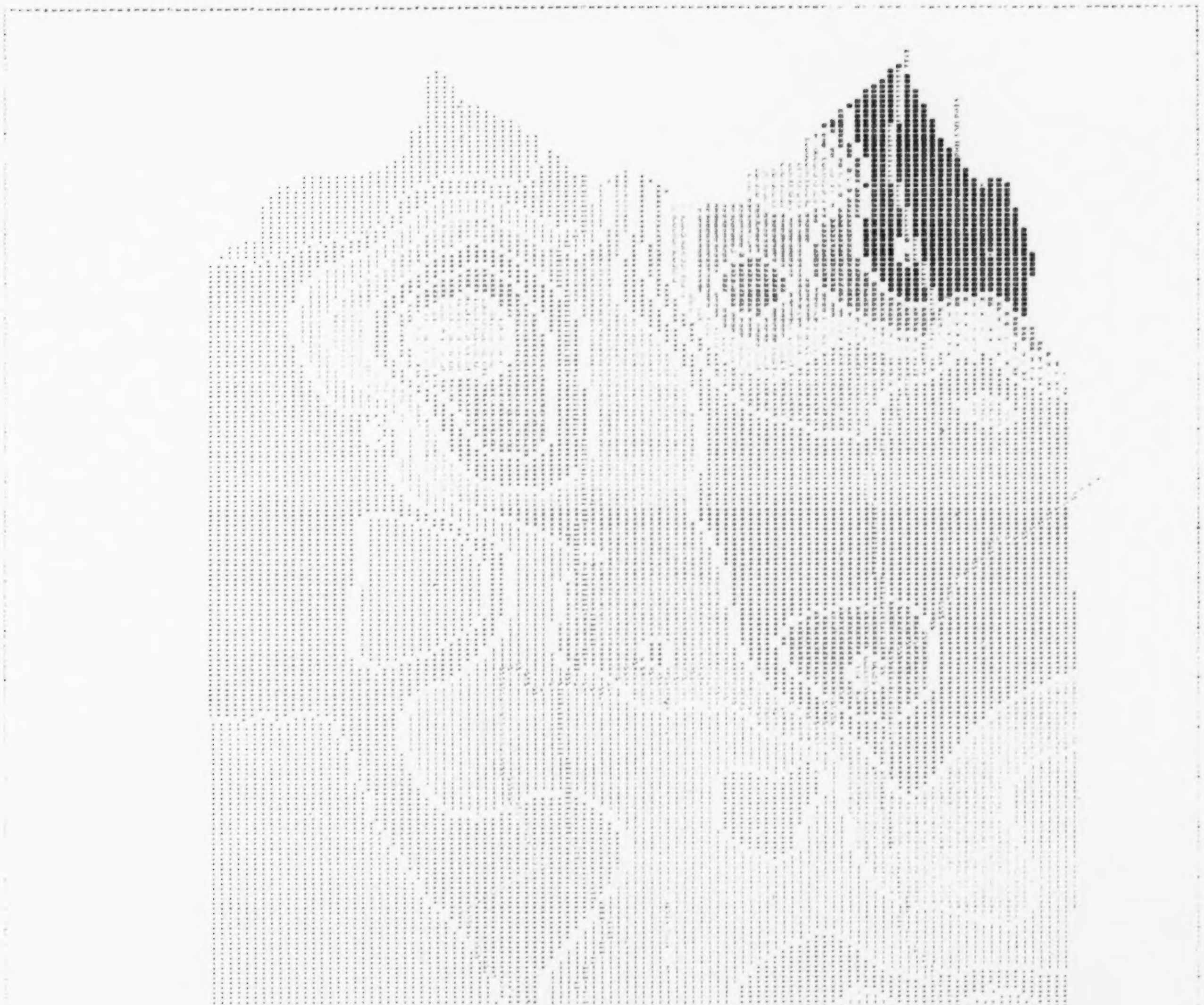
000 0047

```
E-MAP
-----

C    PENNSYLVANIA POPULATION DENSITY

C    PERSONS PER SQUARE MILE


ELECTIVE

   1 MAP SIZE IS          24.00 INCHES LONG BY          36.00 INCHES WIDE
   3 NUMBER OF LEVELS IS 10
   6 LEVEL SIZES ARE PROPORTIONED TO   1)     25.00   2)     25.00   3)     50.00   4)    100.00   5)    100.00
                                       6)    100.00   7)    100.00   8)    100.00   9)    400.00  10)  14584.00
  10 MAP TEXT
                           1965 DATA
                           C. F. GERLACH
  13 MAP SCALE IS      0.11
  14 MARGIN SHIFTS ARE      1.50 AT TOP,      1.50 AT LEFT,      2.00 AT BOTTOM, AND      1.50 AT RIGHT
  15 NEW NUMBER OF LINES PER INCH IS   6.00

TIME = 18:13:13.12

ELAPSE TIME =     3.37 SECONDS FOR INPUT
```

continued

```
MAP   1
-------

C   PENNSYLVANIA POPULATION DENSITY
C   PERSONS PER SQUARE MILE




MAP SCALE =      0.1100 INCHES ON OUTPUT MAP/UNITS ON SOURCE MAP

MAP SHOULD BE PRINTED AT   6.0 ROWS PER INCH AND  10.0 COLUMNS PER INCH

ROW    = (DOWN   COORDINATE -       -9.82) *      0.6600
COLUMN = (ACROSS COORDINATE -       -3.14) *      1.1000


DATA POINTS FOR MAP

POINT        ROW    COLUMN      DATUM      VALUE       LEVEL


   1)         18        43          1    309.00           5
   2)         32        94          2     50.00           2
   3)         34       134          3     55.00           2
   4)         36       157          4     15.00           1
   5)         37       198          5     32.00           1
   6)         36       248          6     48.00           2
   7)         33       280          7     40.00           2
   8)         44       315          8     38.00           2
   9)         55       340          9     17.00           1
  10)         53       293         10    517.00           8
                      .
                      .
                      .
  61)        101       278         61    320.00           6
  62)         88       304         62    654.00           9
  63)        102       325         63    503.00           7
  64)        110       313         64   1042.00          10
  65)        117       298         65    277.00           5
  66)        119       310         66   3005.00          10
  67)        118       324         67  15584.00          10

STANDARD SEARCH RADIUS IS          42.9321

TIME = 18:13:13.59

ELAPSE TIME =     0.47 SECONDS FOR INITIAL CALCULATIONS
```

continued



continued

```
TIME = 18:15:50.71

ELAPSE TIME =  157.12 SECONDS FOR MAP


C   PENNSYLVANIA POPULATION DENSITY

C   PERSONS PER SQUARE MILE


                                                  1965 DATA
                                                  C. F. GERLACH


DATA VALUE EXTREMES ARE         11.00    15584.00


ABSOLUTE VALUE RANGE APPLYING TO EACH LEVEL
      ('MAXIMUM' INCLUDED IN HIGHEST LEVEL ONLY)

  MINIMUM     11.00      35.96      60.96     110.93     210.86     310.79     410.72     510.65     610.58    1010.29
  MAXIMUM     35.93      60.96     110.93     210.86     310.79     410.72     510.65     610.58    1010.29   15584.00


PERCENTAGE OF TOTAL ABSOLUTE VALUE RANGE APPLYING TO EACH LEVEL

              0.16       0.16       0.32       0.64       0.64       0.64       0.64       0.64       2.57      93.58


FREQUENCY DISTRIBUTION OF DATA POINT VALUES IN EACH LEVEL
  LEVEL        1          2          3          4          5          6          7          8          9         10



       ==========================================================================================================
        .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  ØØØØØØØØØ  █████████
        .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  ØØØØØØØØØ  █████████
SYMBOLS ....1....  ''''2''''  ----3----  ====4====  ++++5++++  XXXX6XXXX  OOOO7OOOO  888888888  ØØØØ9ØØØØ  ████*████
        .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  ØØØØØØØØØ  █████████
        .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  ØØØØØØØØØ  █████████
       ==========================================================================================================
FREQ.       7         12         14          8         13          3          3          2          1          4
       1 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI    I88888I    IØØ9ØØI    I██*██I
       2 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI    I88888I               I██*██I
       3 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI                          I██*██I
       4 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I                                                I██*██I
       5 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I
       6 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I
       7 I..1..I    I''2''I    I--3--I    I==4==I    I++5++I
       8            I''2''I    I--3--I    I==4==I    I++5++I
       9            I''2''I    I--3--I               I++5++I
      10            I''2''I    I--3--I               I++5++I
      11            I''2''I    I--3--I               I++5++I
      12            I''2''I    I--3--I               I++5++I
      13                       I--3--I               I++5++I
      14                       I--3--I

TIME = 18:15:52.66

ELAPSE TIME =     1.95 SECONDS FOR HISTOGRAM
```

SAMPLE OUTPUT—Showing Population Density by Counties

```
A-CONFORMOLINES
---------------


VERTEX              DOWN     ACROSS

    1      AREA


(    1)             8.00      52.00
(    2)            27.00      51.00
(    3)            27.00      59.00
(    4)            37.00      59.00
(    5)            36.00      12.00
(    6)            27.00      12.00
(    7)            23.00      23.00
(    8)             8.00      52.00
LENGTH=     137.39
AREA=     792.50
CENTER=(      26.95,      37.26)
                 .
                 .
                 .

   67      AREA


(    1)           156.00     298.00
(    2)           158.00     301.00
(    3)           161.00     301.00
(    4)           168.00     293.00
(    5)           172.00     293.00
(    6)           175.00     286.00
(    7)           170.00     286.00
(    8)           167.00     285.00
(    9)           165.00     289.00
(   10)           161.00     285.00
(   11)           159.00     287.00
(   12)           160.00     289.00
(   13)           160.00     290.00
(   14)           162.00     294.00
(   15)           156.00     298.00
LENGTH=       64.89
AREA=     137.00
CENTER=(     165.09,     291.93)

B-DATA POINTS
-------------


POINT               DOWN     ACROSS

(    1)            18.00      36.00
(    2)            39.00      82.00
(    3)            42.00     119.00
(    4)            44.00     140.00
(    5)            46.00     177.00
(    6)            45.00     222.00
(    7)            40.00     251.00
(    8)            57.00     283.00
(    9)            73.00     306.00
(   10)            70.00     263.00
                .
                .
                .
(   60)           164.00     231.00
(   61)           143.00     250.00
(   62)           124.00     273.00
(   63)           144.00     292.00
(   64)           157.00     281.00
(   65)           158.00     268.00
(   66)           171.00     279.00
(   67)           169.00     291.00
```

continued

```
C-DTCLEGENDS
------------


VERTEX              DOWN    ACROSS     +ROWS     +COLS

(   1) 'PHILADELPHIA' ACROSS FROM

                  165.00    297.00        0.        0.

(   2) 'PITTSBURGH' DOWN FROM

                  135.00     34.00       -5.       -4.

(   3) '*   PENN STATE ' ACROSS FROM

                  111.00    149.00        0.        0.

(   4) 'T' ON LINE


(   1)             99.00      8.00
(   2)            126.00     46.00
(   3)            138.00     53.00
(   4)            147.00     54.00
(   5)            157.00     59.00
         .
         .
         .
(  16)            153.00    194.00
(  17)            148.00    238.00
(  18)            161.00    267.00
(  19)            152.00    295.00
(  20)            155.00    316.00
LENGTH=     351.82

(   5) 'I' ON LINE


(   1)             82.00      9.00
(   2)             81.00     43.00
(   3)             84.00     61.00
(   4)             82.00     68.00
(   5)             84.00     83.00
         .
         .
         .
(  20)             92.00    280.00
(  21)             96.00    283.00
(  22)             96.00    293.00
LENGTH=     297.99

(   6) 'S' ON LINE


(   1)             87.00    141.00
(   2)             78.00    148.00
(   3)             74.00    159.00
(   4)             38.00    167.00
(   5)             81.00    186.00
         .
         .
         .
(  26)            163.00    213.00
(  27)            169.00    223.00
(  28)            172.00    227.00
(  29)            176.00    227.00
(  30)            190.00    238.00
LENGTH=     219.26
```

continued

```
E-VALUES
--------


DATUM          VALUE

(    1)       309.00
(    2)        50.00
(    3)        55.00
(    4)        15.00
(    5)        32.00
(    6)        48.00
(    7)        40.00
(    8)        39.00
(    9)        17.00
(   10)       517.00
          .
          .
          .
(   60)       294.00
(   61)       320.00
(   62)       554.00
(   63)       503.00
(   64)      1042.00
(   65)       277.00
(   66)      3005.00
(   67)     15584.00
```

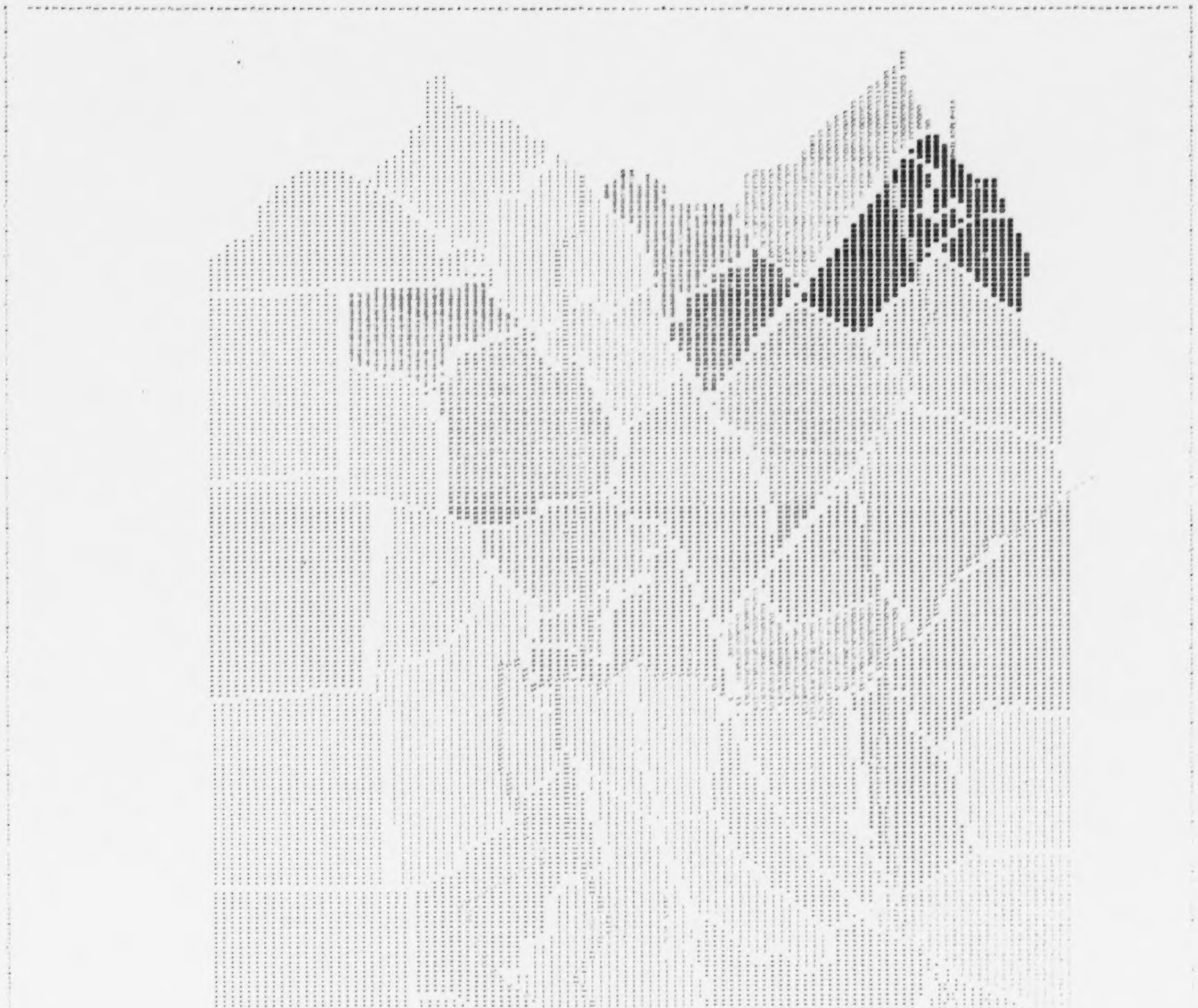
continued

```
E-MAP
-----

C    PENNSYLVANIA POPULATION DENSITY

C    PERSONS PER SQUARE MILE


ELECTIVE


    1 MAP SIZE IS           24.00 INCHES LONG BY           36.00 INCHES WIDE
    2 NUMBER OF LEVELS IS 10
    6 LEVEL SIZES ARE PROPORTIONED TO   1)     25.00   2)     25.00   3)     50.00   4)    100.00   5)    100.00
                                        6)    100.00   7)    100.00   8)    100.00   9)    400.00  10)  14584.00
   10 MAP TEXT
                              1965 DATA
                              C. F. GERLACH
   13 MAP SCALE IS      0.11
   14 MARGIN SHIFTS ARE      1.50 AT TOP,      1.50 AT LEFT,      2.00 AT BOTTOM, AND      1.50 AT RIGHT
   15 NEW NUMBER OF LINES PER INCH IS   6.00

TIME = 18:16:28.95

ELAPSE TIME =   11.56 SECONDS FOR INPUT
```

continued

```
MAP   1
-------

C   PENNSYLVANIA POPULATION DENSITY
C   PERSONS PER SQUARE MILE




MAP SCALE =      0.1100 INCHES ON OUTPUT MAP/UNITS ON SOURCE MAP

MAP SHOULD BE PRINTED AT    6.0 ROWS PER INCH AND  10.0 COLUMNS PER INC

ROW    = (DOWN   COORDINATE -       -9.82) *     0.6600
COLUMN = (ACROSS COORDINATE -       -3.14) *     1.1000


CONFORMOLINES FOR MAP

 ZONE       TYPE       ROW     COLUMN      DATUM     VALUE      LEVEL

     1)       A         21         42         1    309.00          5
     2)       A         33         98         2     50.00          2
     3)       A         34        127         3     55.00          2
     4)       A         37        165         4     15.00          1
     5)       A         36        202         5     32.00          1
     6)       A         36        243         6     48.00          2
     7)       A         33        284         7     40.00          2
     8)       A         42        315         8     38.00          2
     9)       A         54        329         9     17.00          1
    10)       A         52        254        10    517.00          8
                   .
                   .
                   .
    60)       A        115        257        60    294.00          5
    61)       A         98        275        61    320.00          6
    62)       A         88        296        62    654.00          9
    63)       A        100        324        63    503.00          7
    64)       A        106        312        64   1042.00         10
    65)       A        117        288        65    277.00          5
    66)       A        119        308        66   3005.00         10
    67)       A        116        326        67  15584.00         10

TIME = 13:16:30.20

ELAPSE TIME =     1.25 SECONDS FOR INITIAL CALCULATIONS
```

continued



continued

```
TIME = 18:19:42.08

ELAPSE TIME =  191.88 SECONDS FOR MAP


C   PENNSYLVANIA POPULATION DENSITY

C   PERSONS PER SQUARE MILE


                                                          1965 DATA
                                                          C. F. GERLACH


DATA VALUE EXTREMES ARE         11.00    15584.00


ABSOLUTE VALUE RANGE APPLYING TO EACH LEVEL
       ('MAXIMUM' INCLUDED IN HIGHEST LEVEL ONLY)

 MINIMUM     11.00     35.98     60.96    110.93    210.86    310.79    410.72    510.65    610.58   1010.29
 MAXIMUM     35.98     60.96    110.93    210.86    310.79    410.72    510.65    610.58   1010.29  15584.00


PERCENTAGE OF TOTAL ABSOLUTE VALUE RANGE APPLYING TO EACH LEVEL

              0.16      0.16      0.32      0.64      0.64      0.64      0.64      0.64      2.57     93.59


FREQUENCY DISTRIBUTION OF DATA POINT VALUES IN EACH LEVEL
 LEVEL        1         2         3         4         5         6         7         8         9        10



         ==========================================================================================
          .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  XXXXXXXXX  █████████
          .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  XXXXXXXXX  █████████
 SYMBOLS  ....1....  ''''2''''  ----3----  ====4====  ++++5++++  XXXX6XXXX  OOOO7OOOO  888888888  XXXX9XXXX  ████*████
          .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  XXXXXXXXX  █████████
          .........  '''''''''  ---------  =========  +++++++++  XXXXXXXXX  OOOOOOOOO  888888888  XXXXXXXXX  █████████
         ==========================================================================================
 FREQ.       7         12         14          8         13          3          3          2          1          4
       1  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI    I88888I    IXX9XXI    I██*██I
       2  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI    I88888I               I██*██I
       3  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I    IXX6XXI    IOO7OOI                          I██*██I
       4  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I                                                I██*██I
       5  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I
       6  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I
       7  I..1..I   I''2''I    I--3--I    I==4==I    I++5++I
       8            I''2''I    I--3--I    I==4==I    I++5++I
       9            I''2''I    I--3--I               I++5++I
      10            I''2''I    I--3--I               I++5++I
      11            I''2''I    I--3--I               I++5++I
      12            I''2''I    I--3--I               I++5++I
      13                       I--3--I               I++5++I
      14                       I--3--I

TIME = 18:19:42.84

ELAPSE TIME =    0.76 SECONDS FOR HISTOGRAM
```

COST ESTIMATE

For the jobs listed on the Sample Input, the total running time was 118 seconds for the *contour* map and 168 seconds for the *conformant* map. At the current rate for computer time ($0.11/sec.) at The Pennsylvania State University, the chargeable time amounts are the following.

Charge to user = computer time + network overhead
= $12.98 + network overhead—*for the contour map*
= $18.48 + network overhead—*for the conformant map*

CONTENTS—SYMAP

pages	
1– 3	Identification & Abstract
5– 7	User Instructions
9–26	I/O
27	Cost—Contents

DESCRIPTIVE TITLE	Quick-Draw Graphics System
CALLING NAME	QDGS
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Jeff Raskin, The Pennsylvania State Univ. Computation Center; extended and revised by Graham Donaldson, The Pennsylvania State Univ. Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Daniel L. Bernitt, 105 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

QDGS is a set of FORTRAN subroutines that can be used to draw charts, graphs, diagrams, maps, or any other form of pictorial output. The graphic material is produced on the CalComp plotter and other graphic devices as they become available.

To use QDGS, a program is written in FORTRAN IV for the IBM System/360 computer. This program calls the QDGS subroutines; the subroutines produce output (in the form of cards or tape) that is later put into a special system to produce the actual graphical output. When the cards (or tape) are received from the dispatcher and the user has verified that the run was successful, he usually submits the cards to the dispatcher along with a plot request form.

The three concepts underlying the design of QDGS are (1) the description of an elementary picture in terms of coordinates, (2) the geometric transformation of pictures (into various positions and orientations), and (3) the building of complex pictures from elementary ones. There will be subroutines to do each of the three basic processes. In addition, there are special routines to draw axes for graphs and charts, to do lettering and numbering, and automatically to do the necessary task of punching the cards or

continued

writing the tape used in plotting (physically drawing on the plotting device). QDGS is intended as a nucleus about which the user's graphic routines, tailored to his own unique application, can be written.

USER INSTRUCTIONS

Initialization

Before any portion of the QDGS may be used by a program, four arrays must be specified. These arrays are used by the system as a working area and may not be used or altered in any way by the user's program. It is difficult to estimate the size of the arrays as space requirements vary widely with different program; however, the following declarations will do for many jobs. If the message LAVS EXHAUSTED appears, the 1000 may be increased.

```
REAL D(6,1000)
INTEGER * 2 A(1000), B(1000), C(1000)
```

Any FORTRAN variable names may be used for the A, B, C, and D. The first subscript of the D array must be 6. Once the arrays have been initialized, and before any other QDGS subroutines are called, the following statement must be used

```
CALL INITQ(A,B,C,D,1000)
```

Defining Elementary Pictures

Pictures are defined in Cartesian coordinates; however, with each point (x,y), we now associate a third coordinate z, which has a code for the condition of the plotter pen in *arriving* at the point (x,y). The convention used is to code z=0 to indicate "without drawing" and z=1 to indicate "drawing." The desired picture is achieved, therefore, by storing these ordered triples. This is the purpose of the ADPTQ subroutine (see Summary of Subroutines).

Building Pictures from Pictures

In the QDGS, pictures are formed out of line segments defined with ADPTQ and by transformed versions of pictures previously defined. A geometric transformation is applied to the previously defined picture before being placed into another picture. There are six basic transformations (and compounds of these) allowed by the QDGS:

(1) translation in X (TX)—motion of a figure horizontally

(2) translation in Y (TY)—motion of a figure vertically

(3) scaling in S (SX)—horizontal expansion or contraction

(4) scaling in Y (SY)—vertical expansion or contraction

(5) rotation (TH)—the TH is for theta, traditonally used to denote an angle—here measured *counterclockwise* in radians

continued

(6) skew (or shear) (SK)—a tilting that leaves height unchanged, measure in radians clockwise from the positive Y direction

Transformations are represented by variables and must, therefore, be dimensioned (using 6 as the dimension); their values, however, are set by calling the subroutine XFSTQ. Compound and inverse transformations may be performed by using subroutines XFXFQ and XFINQ, respectively. Adding pictures (building more complex ones) can be achieved by using ADELQ, which also allows a picture to be transformed before being added to another picture.

Displaying a Completed Picture

Once a picture has been created and given a name, the card or tape output representing it may be obtained. Up to now, pictures have been described in terms of abstract coordinates, but the *physical* size is determined by a physical scale factor in the call to the outputting routine, subroutine DISPQ. The scale factor itself, however, is determined by the desired units of the user and the plotter's raster unit. (A raster unit is the smallest distance over which a line can be drawn by a particular device.)

Axes

A single subroutine, AXISQ, provides a logarithmic or linear axis with appropriate labeling of the tic marks as an option.

Drawing Labels and Titles

A legend on a graph or chart, the text that accompanies a drawing, sizes, tolerance, and the like are produced by subroutine LABLQ (for alphamerics) and NMBRQ (for numbers). A label is itself a picture and differs from other pictures only in that a subroutine sets up the picture automatically. The user transforms it to the desired size and position by using the same transformations as for any other picture.

REFERENCES

Primer on the Quick-Draw Graphics System (QDGS), Pa. State Univ. Computation Ctr. (July 1967).

Summary of Subroutines

Name	Form of Call	Meaning of Parameters	Purpose and/or Abilities
INITQ	CALL INITQ (A,B,C,D, 1000)	any FORTRAN variables names for A,B, C,D	initialization (required after declaring A,B,C,D)
ADPTQ	CALL ADPTQ (X,Y,Z, NAME)	X,Y: Cartesian coordinates; Z: plotter per code NAME: name for picture	adds (stores) points to an elementary picture
XFSTQ	CALL XFSTQ (TX,TY,SX, SY,TH,SK,Q)	first six already described; Q: transformation to be given the value	transforming routine (transformations performed in order SK,TH,SY,SX,TY,TX)
XFXFQ	CALL XFXFQ (S,Q,T)	Q,S: two predfined transformations T: the new compound transformation of Q and S	allows compounding of transformations (Q performed first, then S)
XFINQ	CALL XFINQ (Q,QI)	Q: predefined transformation QI: new, inverse transformations	creates inverse transformations
ADELQ	CALL ADELQ (N,Q,M)	N: picture to be added Q: transformation to be applied to N M: new picture or picture to be built up	allows pictures of any structure depth to be created
DISPQ	CALL DISPQ (N,SCLF)	N: name of picture to be output SCLF: scale factor for plotting	actual displaying routine; for various plotting devices

continued

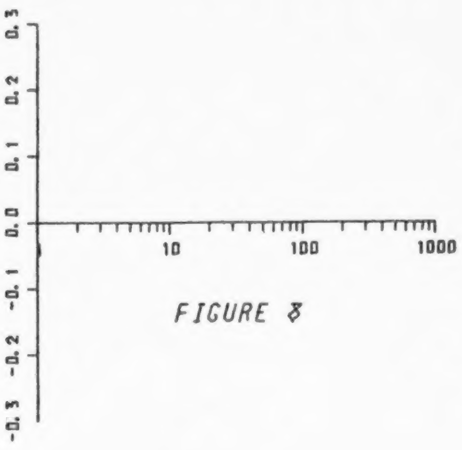
Summary of Subroutines *(cont.)*

Name	Form of Call	Meaning of Parameters		Purpose and/or Abilities
AXISQ	CALL AXISQ (SIZE, K, TMIN, TINC,HITE,N, Q,M)	SIZE: K: TMIN: TINC: HITE: N: Q: M:	overall axis length number of segments value placed at first mark increment for each succeeding mark size of axis marks and numbers appearing with the axis number of digits to right of decimal point in numbers transformation to be applied to axis name of axis pictured	Sp,e options may be indicated by using each parameter. Minus signs before the values of SIZE,K, and HITE indicate axis suppression, logarithmic (rather than linear) scaling, and placement of numbers on the *other* side of an axis. If K is negative, TINC represents the constant ratio between marks rather than the linear increment; if N=-2, numbering is suppressed for the axis (TMIN and TINC are ignored)
LABLQ	CALL LABLQ (LABEL, LGTH,Q,M)	LABEL: LGTH: Q: M:	label (in quotes) number of characters transformation to be applied picture to which label is to be added	LABEL may be replaced by a variable representing alphanumeric data; a minus sign before LGTH indicates centering rather than left-justification
NMBRQ	CALL NMBRQ (R,N,IC, Q,M)	R: N: IC: Q,M:	real number desired number of digits behind decimal point position code same as above	N=-1 suppresses decimal IC=0 for centering, 1 for left-justification; R may have no more than 8 digits to the left and 7 digits to the right of the decimal point

SAMPLE INPUT WITH CORRESPONDING OUTPUT

EXAMPLE OF A COMPLETE PROGRAM WITH OUTPUT

```
C       PROGRAM TO PRODUCE FIGURE 8 FOR THE QDGS PRIMER
C
C       INITIALIZATION
        INTEGER*2 A(1000),B(1000),C(1000)
        INTEGER M/0/
        REAL D(6,1000),Q(6),PI/3.141593/
        CALL INITQ(A,B,C,D,1000)
C
C       CREATE TRANSFORM FOR Y-AXIS
        CALL XFSTQ(0.0,-1.5,1.00,1.00,PI/2.0,0.0,Q)
C       PUT THE AXIS ONTO PICTURE M
        CALL AXISQ(3.0,6,-0.3,0.1,-0.08,1,Q,M)
C
C       CREATE TRANSFORM FOR THE X-AXIS AND ADD AXIS TO M
        CALL XFSTQ(0.0,0.0,1.00,1.00,0.0,0.0,Q)
        CALL AXISQ(3.0.-3,1.0,10.0,0.08,-1,Q,M)
C
C       PREPARE A TRANSFORM FOR THE LABEL
        CALL XFSTQ(1.5,-.75,.12,.13,0.,.3,Q)
C       ADD THE LABEL TO M
        CALL LABLQ('FIGURE 8',-8,Q,M)
C
C       DISPLAY THE COMPLETED FIGURE 8, SCALED FOR THE CALCOMP
        CALL DISPQ(M,200.0)
C
C       NOTE:  ALL OF THE ILLUSTRATIONS FOR THE QDGS PRIMER WERE
C       DRAWN ON THE CALCOMP PLOTTER.  ONLY THOSE GRAPHIC
C       SUBROUTINES MENTIONED IN THIS PRIMER WERE USED.
        STOP
        END
```



continued

```
C          PROGRAM TO READ POINTS AND PLOT A GRAPH
C
C          ASSUME THE RANGES OF X AND Y ARE KNOWN
C          X: -5 TO +5       Y: -2 TO +2
C          AND THE (X,Y) PAIRS HAVE BEEN SORTED IN ORDER OF INCREASING X.
C          CALL THE GRAPH "GRAPH"
       INTEGER GRAPH/0/,PLUS/0/,INPUT/5/
C          INITIALIZE
       INTEGER*2 AA(1000),BB(1000),CC(1000)
       REAL DD(6,1000),PI/3.141593/,Q(6)
       CALL INITQ(AA,BB,CC,DD,1000)
C
C          PUT X-AXIS ON GRAPH
C          MAKE AXIS RUN FROM -5 TO +5 WITH 10 SEGMENTS, THUS 'SIZE' IS
C          10. AND 'K' IS 10. AXISO PROVIDES AN AXIS STARTING AT 0 AND
C          STRETCHING IN THE POSITIVE X-DIRECTION. THEREFORE, SET TX=-5.
C          IN ORDER TO START THIS AXIS AT -5. NUMBER EACH TIC MARK
C          STARTING WITH -5 AT THE LEFT END AND INCREMENTING BY 1.
C          SUPPRESS THE DECIMAL POINT('N'=-1) AND MAKE THE DIGITS .1 UNIT
C          HIGH('HITE'=.1).
       CALL XFSTQ(-5.,0.,1.,1.,0.,0.,Q)
       CALL AXISQ(10.,10 , -5.,1.,.1,-1,Q,GRAPH)
C          LABEL X-AXIS
C          MAKE LETTERS .1 HIGH TO MATCH NUMBERS. USE N=-10 TO CENTRE
C          LABEL AT (4.,-.5)
       CALL XFSTQ(4 ,-.5,.1,.1,0.,0.,Q)
       CALL LABLO('X-VARIABLE',-10,Q,GRAPH)
C          PUT Y-AXIS ON GRAPH
C          USE 'SIZE'=4. THIS GIVES AN AXIS FROM 0 TO 4 IN X-DIRECTION.
C          ROTATE BY PI/2 FOR AXIS FROM 0 TO 4  IN Y-DIRECTION. TRANSLATE
C          BY -2. FOR AXIS FROM -2 TO +2 IN Y-DIRECTION. NUMBER TIC MARKS
C          AS BEFORE BUT USE 'HITE'=-.1 TO PUT TIC MARKS AND NUMBERS
C          ON THE OTHER SIDE.
       CALL XFSTQ(0.,-2.,1.,1.,PI/2.,0.,Q)
       CALL AXISQ(4.,4,-2.,1.,-.1,-1,Q,GRAPH)
C          LABEL Y-AXIS
C          MAKE LETTERS .1 HIGH TO MATCH NUMBERS. USE N=-10 TO CENTRE
C          LABEL AT (-.4,1.)
       CALL XFSTQ(-.4,1.,.1,.1,PI/2.,0.,Q)
       CALL LABLO('Y-VARIABLE',-10,Q,GRAPH)
C          IF DATA POINTS ARE TO BE DENOTED BY A SPECIAL SYMBOL,
C          CREATE IT NOW,E.G.,A PLUS SIGN CALLED "PLUS"
C          MAKE IT SO THAT THE PEN IS BACK AT THE STARTING POINT AFTER
C          DRAWING IT. ALSO MAKE IT A STANDARD SIZE(1 BY 1).
       CALL ADPTQ(-.5,0.,0,PLUS)
       CALL ADPTQ(.5,0.,1,PLUS)
       CALL ADPTQ(0.,-.5,0,PLUS)
       CALL ADPTQ(0.,.5,1,PLUS)
       CALL ADPTQ(0.,0.,0,PLUS)
C
C          MOVE PEN TO FIRST DATA POINT WITHOUT DRAWING A LINE.
       READ(INPUT,2,END=4) X,Y
     2 FORMAT(2F10.2)
       CALL ADPTO(X,Y,O,GRAPH)
C          OMIT NEXT SET OF STATEMENTS IF NO SPECIAL SYMBOL IS TO BE
C          PLACED AT THIS POINT.
C          MAKE PLUS SIGN .1 UNIT SQUARE ON GRAPH,PLACED AT (X,Y)
       CALL XFSTQ(X,Y,.1,.1,0.,0.,Q)
       CALL ADELQ(PLUS,Q,GRAPH)
C          READ DATA POINTS AND PUT ON GRAPH
     3 READ(INPUT,2,END=4) X,Y
C          OMIT NEXT STATEMENT IF POINTS ARE NOT TO BE JOINED BY A LINE
       CALL ADPTQ(X,Y,1,GRAPH)
C          OMIT NEXT SET OF STATEMENTS IF NO SPECIAL SYMBOL IS TO BE
C          PLACED AT EACH POINT.
       CALL XFSTQ(X,Y,.1,.1,0.,0.,Q)
       CALL ADELQ(PLUS,Q,GRAPH)
C          (X,Y) HAS NOW BEEN ADDED TO GRAPH
       GO TO 3
C
     4 CONTINUE
C          GRAPH IS COMPLETED. CHOOSE SCALEFACTOR AND DISPLAY IT.
       CALL DISPO(GRAPH,200.)
       STOP
       END
```

COST ESTIMATE

For the jobs listed as the Sample Input, no exact timing figures were available. However, none of the included examples would require more than 15 seconds of computer time to run. At the current rate of charge at The Pennsylvania State University ($0.11/sec.), this amounts to a maximum of $1.65 per example.

Approximate charge to user = $1.65 (maximum) X number of examples
+ network overhead

CONTENTS—QDGS

pages

1– 2 Identification & Abstract
3– 6 User Instructions
7– 8 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Key Grapheme in Context
CALLING NAME	KGIC
INSTALLATION NAME	Washington University Computing Facilities
AUTHOR(S) AND AFFILIATION(S)	Joel Achtenberg Computing Facilities Washington University
LANGUAGE	PL/I
COMPUTER	S/360 under OS
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	J. Philip Miller, Computing Facilities, Washington University, St. Louis, Mo. 63130 Tel.: (314) 863-0100 ext. 4041

FUNCTIONAL ABSTRACT

The KGIC program was written to facilitate the analysis of the environmental distribution of graphic characters. It produces a KWIC-like listing of all occurrences of a given grapheme along with the graphic environment in which each instance appeared. The listing may be sorted either forward or backward from the key grapheme to facilitate inspection. Provision is made for specification by the user of special alphabets for foreign languages or for phonemic transcriptions.

If any word appears more than once in the data to be processed, only one set or records will be produced for that word. A counter will be increased and the frequency of occurrence printed in both the alphabetical listing and in the KGIC listing. The alpha listing thus will contain a complete frequency count of the corpus under consideration.

It should be noted that whenever the user can attribute phonetic or phonemic status to individual graphemes, the KGIC listing provides correspondingly significant information about phonetic and/or phonemic environments.

continued

The program produces the items listed below as output.

1. an alphabetical listing of all words processed by the KGIC program, along with their absolute and relative frequencies of occurrence,

2. the KGIC listing itself grouped by alphabetical character, with the absolute frequency of occurrence given for each unique occurrence, and the total number of occurrences and the total number of unique occurrences given for each character,

3. a summary table containing the absolute and relative frequencies of occurrence for both the total number of occurrences and for the total of unique occurences,

4. optionally, a horizontal bar graph of the relative frequencies of all occurrences of each grapheme,

5. a number of summary statistics, i.e.,

   a) total number of words processed, i.e., tokens
   b) average length of word
   c) total number of unique words, i.e., types
   d) the type/token ratio
   e) total number of characters processed
   f) total number of unique occurrences of all characters

6. a statement of all program options used in a particular run, and a complete listing of the EMICTT, defining the alphabet in use for that run.

USER INSTRUCTIONS

A number of the run-time options are available. One or more options may be selected by including the appropriate information in the data set OPTIN. For each available option, except TITLE, a default is indicated which will apply if the given option is not specified. Two or more options are separated by commas, and the final option is followed by a semicolon.

1. TITLE

The user must specify a project title to be used in the headings of certain parts of the program output. The title must be no longer than 80 characters, including blanks, and must be enclosed within single quote marks, as follows,

TITLE = 'users project title'

2. SORT

As indicated previously, the KGIC listing can be sorted either forward or backward from the key character. Merely code BACKWARD or FORWARD, as follows,

```
SORT = ‾'BACKWARD'‾
       _'FORWARD' _
```

If SORT is not specified, the list is alphabetized forward by default.

3. GRAPHX

A horizontal bar graph displaying graphically the frequency of occurrence of each grapheme which appeared in the data may be produced. Code,

```
GRAPHX ='YES'
        'NO'
```

The default option bypasses graphing.

4. SPACING

To indicate desired spacing of the alpha and KGIC listings, merely code

SPACING = n

where n is the number of lines skipped after printing. Default is 1 (i.e., single spacing).

continued

5. EMICTT

Because the program is designed for graphic analysis of foreign languages as well as English, provision has been made for the specification of special alphabets designed by the user for foreign languages or for phonetic transcription. Such specification is done through the Expanded Master Input Character Translate Table (EMICTT), which consists of two major parts. The first, MICTT, specifies the use to be made of each of the 256 characters recognized by the 360. The codes employed are as follows.

1 = Ignore, not used as print.

2 = Delimiter for ID fields, comments, titles, and other information to be ignored in processing.

3 = Space.

4 = Character to be considered part of the alphabet in use.

If any character other than 1 to 4 is coded, the improper code will be replaced by code 4, and an error message produced.

The second part of EMICTT specifies the character(s) to be produced on output for each character designated as part of the alphabet (i.e., those coded '4'). There are two possibilities. First, each character can be represented by a single character from the S/360 TN train, or other available print train. Second, for additional flexibility, two characters may be overprinted to yield a composite character. For example, the break character (_') might be used to distinguish low toned vowels from other vowels. In either case, printing is determined by specification of the appropriate character in either PRT(1), for base characters, or PRT(2), for overprinted characters.

For both MICTT and PRT(1) and PRT(2), default values specified are for the standard English (ROMAN) alphabet, as given in Appendix A. Changes are made in MICTT by coding as follows,

```
MICTT (xxx) = n
```

continued

where xxx is the decimal number from 1 to 255 specifying the internal character under consideration, and n is the usage code (1–4). Changes are made in the print routine by coding

```
PRT(i, xxx) = 'c'
```

where xxx is the decimal number from 1 to 255, i designates the portion of PRT involved (i.e., 1 = base character, 2 = overprint), and c is the punched code for the desired print character.

Note that hex zero is a reserved character used in the production of the alpha listing. Attempts to change MICTT (0) or PRT(i,0) will be ignored and an error message produced.

Multiple changes in PRT and MICTT can be made by separating them by commas.

For example, to treat the hyphen (-) and the apostrophe (') as characters to be ignored rather than part of the alphabet, code as follows,

```
MICTT(96)  = 1, PRT(1,96) = '',
MICTT(125) = 1, PRT(1,125) = '',
```

A modification of EMICTT for use with a phonemic transcription of the Mayan Language is shown in Appendix B. Upper-case consonants are used for glottalized sounds. High-, mid-, and low-toned vowels are represented by upper-case, lower-case and underlined lower-case vowels respectively.

Description of Input

Input consists of two files, one carrying the run-time program options (OPTIN), and the other carrying the data itself (SYSIN).

File OPTIN consists of one or more cards containing the user-specified project title and one or more program options, or none at all. The title and each of the program options are separated from each other by commas, and the final option is followed by a semicolon. If no program options are specified the title must be followed by a semicolon.

continued

A typical example might be as follows,

```
//GO.OPTIN DD *
TITLE = 'MAYAN PROJECT', SPACING = 2
GRAPHX = 'YES';
/*
```

File SYSIN carries the data to be analyzed. The input stream may contain any acceptable 8-bit code except 0000000 (hex 00) which is reserved for internal use. For example,

```
//GO.SYSIN DD *

THIS IS A SAMPLE OF INPUT DATA.  DEFAULT VALUES OF
MICTT WILL IGNORE PUNCTUATION AND BYPASS ALL CHARAC-
TERS BETWEEN PARENTHESES.  FOR EXAMPLE (TITLE IS
"SAMPLE DATA") WILL BE IGNORED.

/*
```

Running Instructions

The KGIC load module is contained in a partitioned data set DS09810.JALIB, stored in disk, and can thus be executed directly without compilation. In addition to the data input file, SYSIN, and the program options file, OPTIN, the program requires three work files for the SORT routine and the file SYSPRINT which contains the program output.

The JCL for a typical run is illustrated below. This is only an illustration, however, and must be adapted to the user's specifications.

```
---current OS JOB card---
//JOBLIB DD DSNAME=DS09810.JALIB,DISP=SHR
//  EXEC  PGM=KGIC
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,500,,CONTIG)
//SORTWK02 DD UNIT=(SYSDA,SEP=SORTWK01),SPACE=(TRK,500,,CONTIG)
//SORTWK03 DD UNIT=(SYSDA,SEP=(SORTWK02,SORTWK01)),
//                SPACE=(TRK,500,,CONTIG)
//SYSOUT DD SYSOUT=A
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSLIN DD UNIT=SYSDA,SPACE=(TRK,(10,10))
//SYSUTI DD UNIT=SYSDA,SPACE=(TRK,(10,10))
//SYSPRINT DD SYSOUT=A   CONTAINS THE PROGRAM OUTPUT
//SYSIN    DD UNIT=TAPE9,DISP=(OLD,KEEP) CONTAINS INPUT DATA
//OPTIN    DD *
---One of more cards with program options and project title---
/*
```

continued

Appendix A: Default values for EMICTT

EXPANDED MASTER INPUT CHARACTER TRANSLATE TABLE

DEC	CODE	PRT(1)	PRT(2)
0	1		
1	1		
2	1		
3	1		
4	1		
5	1		
6	1		
7	1		
8	1		
9	1		
10	1		
11	1		
12	1		
13	1		
14	1		
15	1		
16	1		
17	1		
18	1		
19	1		
20	1		
21	1		
22	1		
23	1		
24	1		
25	1		
26	1		
27	1		
28	1		
29	1		
30	1		
31	1		
32	1		
33	1		
34	1		
35	1		
36	1		
37	1		
38	1		
39	1		
40	1		
41	1		
42	1		
43	1		
44	1		
45	1		
46	1		
47	1		
48	1		
49	1		
50	1		
51	1		
52	1		
53	1		
54	1		
55	1		
56	1		
57	1		
58	1		
59	1		
60	1		
61	1		
62	1		
63	1		
64	3		
65	1		
66	1		
67	1		
68	1		
69	1		
70	1		
71	1		
72	1		
73	1		
74	1		
75	1		
76	2		
77	1		
78	1		
79	1		
80	1		

continued

DEC	CODE	PRT(1)	PRT(2)	DEC	CODE	PRT(1)	PRT(2)
81	1			126	1		
82	1			127	1		
83	1			128	1		
84	1			129	4	a	
85	1			130	4	b	
86	1			131	4	c	
87	1			132	4	d	
88	1			133	4	e	
89	1			134	4	f	
90	1			135	4	g	
91	2			136	4	h	
92	1			137	4	i	
93	1			138	1		
94	1			139	2		
95	1			140	1		
96	4	-		141	1		
97	1			142	1		
98	1			143	1		
99	1			144	1		
100	1			145	4	j	
101	1			146	4	k	
102	1			147	4	l	
103	1			148	4	m	
104	1			149	4	n	
105	1			150	4	o	
106	1			151	4	p	
107	1			152	4	q	
108	1			153	4	r	
109	4	_		154	1		
110	2			155	2		
111	1			156	1		
112	1			157	1		
113	1			158	1		
114	1			159	1		
115	1			160	1		
116	1			161	1		
117	1			162	4	s	
118	1			163	4	t	
119	1			164	4	u	
120	1			165	4	v	
121	1			166	4	w	
122	1			167	4	x	
123	1			168	4	y	
124	1			169	4	z	
125	4	'		170	1		

continued

DEC	CODE	PRT(1)	PRT(2)
171	1		
172	1		
173	1		
174	1		
175	1		
176	1		
177	1		
178	1		
179	1		
180	1		
181	1		
182	1		
183	1		
184	1		
185	1		
186	1		
187	1		
188	1		
189	1		
190	1		
191	1		
192	1		
193	4	A	
194	4	B	
195	4	C	
196	4	D	
197	4	E	
198	4	F	
199	4	G	
200	4	H	
201	4	I	
202	1		
203	1		
204	1		
205	1		
206	1		
207	1		
208	1		
209	4	J	
210	4	K	
211	4	L	
212	4	M	
213	4	N	
214	4	O	
215	4	P	
216	4	Q	
217	4	R	
218	1		
219	1		
220	1		
221	1		
222	1		
223	1		
224	1		
225	1		
226	4	S	
227	4	T	
228	4	U	
229	4	V	
230	4	W	
231	4	X	
232	4	Y	
233	4	Z	
234	1		
235	1		
236	1		
237	1		
238	1		
239	1		
240	4	0	
241	4	1	
242	4	2	
243	4	3	
244	4	4	
245	4	5	
246	4	6	
247	4	7	
248	4	8	
249	4	9	
250	1		
251	1		
252	1		
253	1		
254	1		
255	1		

continued

Appendix B: EMICTT for a Mayan Language Project

EXPANDED MASTER INPUT CHARACTER TRANSLATE TABLE

DEC	CODE	PRT(1)	PRT(2)
0	1		
1	4	'	
2	4	b	
3	4	z	
4	4	Z	
5	4	c	
6	4	C	
7	4	h	
8	4	k	
9	4	K	
10	4	1	
11	4	m	
12	4	n	
13	4	p	
14	4	P	
15	4	s	
16	4	x	
17	4	t	
18	4	T	
19	4	w	
20	4	y	
21	4	A	
22	4	a	
23	4	a	–
24	4	E	
25	4	e	
26	4	e	–
27	4	I	
28	4	i	
29	4	i	–
30	4	0	
31	4	o	
32	4	o	–
33	4	U	
34	4	u	
35	4	u	–

Characters 36-255 are not used and are coded 1, with PRT(1) and PRT(2) null.

SAMPLE INPUT

Input for the sample program consists of a data stream that is not keypunchable. However, it is similar in content to the sample given in the Description of Input section of the User Instructions.

SAMPLE OUTPUT

```
ALPHA LISTING OF INPUT ITEMS



         A
         ALL
         AND
         BE
         BETWEEN
         CAUSE
         CHARACTERS
         DATA
         DEFAULT
         DELIMITERS
         EXAMPLE
         FOR
         IGNORED
         INPUT
         IS
         KGIC
         MARKS
         MICTT
         OF
         PARAGRAPH
         PROGRAM
         PROVIDED
         PUNCTUATION
         SAMPLE
         TABLE
         THE
         THIS
         TITLE
         TO
         VALUES
         WILL


TYPES=         31

TOKENS=         43
```

```
KGIC: KEY GRAPHEME IN CONTEXT LISTING

  A TYPICAL KGIC RUN WITH ENGLISH DATA


>>>>>>>>>>>>>>>>>>>>>>>>A<<<<<<<<<<<<<<<<<<<<<<<<

                        A                                   1
                     DATA                                   2
                       TABLE                                1
                    CHARACTERS                              1
                     PARAGRAPH                              1
                        ALL                                 2
                       VALUES                               1
                   PROGRAM                                  1
                       SAMPLE                               2
                      EXAMPLE                               1
                        AND                                 1
                  PARAGRAPH                                 1
                      CHARACTERS                            1
                       PARAGRAPH                            1
                       MARKS                                1
                       DATA                                 2
                  PUNCTUATION                               1
                     DEFAULT                                1
                       CAUSE                                1
THERE ARE A TOTAL OF     23 OCCURRANCES OF GRAPHEME A
THERE ARE     19 UNIQUE OCCURRANCES OF GRAPHEME A
```

continued

049-13

```
0
1
2
1
2
1
1
1
2
1
1
1
2
2
1
2
1
1
1
2
1
1
1
1
2
1
2
2
1
2
1
2
```

```
>>>>>>>>>>>>>>>>>>>>>>>>B<<<<<<<<<<<<<<<<<<<<<<<<

                         BE                                     2
                         BETWEEN                                1
                       TABLE                                    1
THERE ARE A TOTAL OF       4 OCCURRANCES OF GRAPHEME b
THERE ARE       3 UNIQUE OCCURRANCES OF GRAPHEME B

>>>>>>>>>>>>>>>>>>>>>>>>C<<<<<<<<<<<<<<<<<<<<<<<<

                     KGIC                                       1
                        CAUSE                                   1
                        CHARACTERS                              1
                   CHARACTERS                                   1
                       MICTT                                    1
                     PUNCTUATION                                1
THERE ARE A TOTAL OF       6 OCCURRANCES OF GRAPHEME C
THERE ARE       6 UNIQUE OCCURRANCES OF GRAPHEME C

>>>>>>>>>>>>>>>>>>>>>>>>D<<<<<<<<<<<<<<<<<<<<<<<<

                        AND                                     1
                    IGNORED                                     2
                   PROVIDED                                     1
                        DATA                                    2
                     PROVIDED                                   1
                        DEFAULT                                 1
                        DELIMITERS                              1
THERE ARE A TOTAL OF       9 OCCURRANCES OF GRAPHEME D
THERE ARE       7 UNIQUE OCCURRANCES OF GRAPHEME D

>>>>>>>>>>>>>>>>>>>>>>>>E<<<<<<<<<<<<<<<<<<<<<<<<

                         BE                                     2
                        THE                                     2
                      CAUSE                                     1
                      TABLE                                     1
                      TITLE                                     1
                     SAMPLE                                     2
                    EXAMPLE                                     1
                     IGNORED                                    2
                    PROVIDED                                    1
                      BETWEEN                                   1
                         DEFAULT                                1
                         DELIMITERS                             1
                     BETWEEN                                    1
                   CHARACTERS                                   1
                   DELIMITERS                                   1
                      VALUES                                    1
                         BETWEEN                                1
                          EXAMPLE                               1
THERE ARE A TOTAL OF      22 OCCURRANCES OF GRAPHEME E
THERE ARE      18 UNIQUE OCCURRANCES OF GRAPHEME E

>>>>>>>>>>>>>>>>>>>>>>>>F<<<<<<<<<<<<<<<<<<<<<<<<

                         OF                                     2
                        DEFAULT                                 1
                          FOR                                   2
THERE ARE A TOTAL OF       5 OCCURRANCES OF GRAPHEME F
THERE ARE       3 UNIQUE OCCURRANCES OF GRAPHEME F

>>>>>>>>>>>>>>>>>>>>>>>>G<<<<<<<<<<<<<<<<<<<<<<<<

                         KGIC                                   1
                          IGNORED                               2
                       PROGRAM                                  1
                     PARAGRAPH                                  1
THERE ARE A TOTAL OF       5 OCCURRANCES OF GRAPHEME G
THERE ARE       4 UNIQUE OCCURRANCES OF GRAPHEME G
```

continued

```
>>>>>>>>>>>>>>>>>>>>>>>>>H<<<<<<<<<<<<<<<<<<<<<<<<<

                    PARAGRAPH                                    1
                           CHARACTERS                            1
                           THE                                   2
                           THIS                                  2
THERE ARE A TOTAL OF      6 OCCURRANCES OF GRAPHEME H
THERE ARE      4 UNIQUE OCCURRANCES OF GRAPHEME H
```

A_FEW_STATISTICS

```
TOTAL NUMBER OF WORDS PROCESSED(TOKENS) IS        43

NUMBER OF DIFFERENT WORDS(TYPES) IS       31

TYPE/TOKEN RATIO IS      0.721

TOTAL NUMBER OF CHARACTERS PROCESSED IS       204

TOTAL NUMBER OF UNIQUE OCCURRENCES OF ALL CHARACTERS IS       162

AVERAGE LENGTH OF WORD IS 4.744

FREQUENCIES OF OCCURRANCE OF EACH GRAPHEME

                    ALL_OCCURRENCES      UNIQUE_OCCURRENCES
       GRAPHEME  ABSOLUTE  RELATIVE     ABSOLUTE  RELATIVE

           -           0     0.000            0     0.000
                       0     0.000            0     0.000
           ,           0     0.000            0     0.000
           a           0     0.000            0     0.000
           b           0     0.000            0     0.000
           c           0     0.000            0     0.000
           d           0     0.000            0     0.000
           e           0     0.000            0     0.000
           f           0     0.000            0     0.000
           g           0     0.000            0     0.000
           h           0     0.000            0     0.000
           i           0     0.000            0     0.000
           j           0     0.000            0     0.000
           k           0     0.000            0     0.000
           l           0     0.000            0     0.000
           m           0     0.000            0     0.000
           n           0     0.000            0     0.000
           o           0     0.000            0     0.000
           p           0     0.000            0     0.000
           q           0     0.000            0     0.000
           r           0     0.000            0     0.000
           s           0     0.000            0     0.000
           t           0     0.000            0     0.000
           u           0     0.000            0     0.000
           v           0     0.000            0     0.000
           w           0     0.000            0     0.000
           x           0     0.000            0     0.000
           y           0     0.000            0     0.000
           z           0     0.000            0     0.000
           A          23     0.113           19     0.117
           B           4     0.020            3     0.019
           C           6     0.029            6     0.037
           D           9     0.044            7     0.043
           E          22     0.108           18     0.111
           F           5     0.025            3     0.019
           G           5     0.025            4     0.025
           H           6     0.029            4     0.025
           I          16     0.078           12     0.074
           J           0     0.000            0     0.000
           K           2     0.010            2     0.012
```

continued

```
L          16     0.078        11     0.068
M           7     0.034         6     0.037
N           7     0.034         6     0.037
O          11     0.054         7     0.043
P           9     0.044         8     0.049
Q           0     0.000         0     0.000
R          13     0.064        11     0.068
S          11     0.054         8     0.049
T          20     0.098        16     0.099
U           6     0.029         6     0.037
V           2     0.010         2     0.012
W           3     0.015         2     0.012
X           1     0.005         1     0.006
Y           0     0.000         0     0.000
Z           0     0.000         0     0.000
0           0     0.000         0     0.000
1           0     0.000         0     0.000
2           0     0.000         0     0.000
3           0     0.000         0     0.000
4           0     0.000         0     0.000
5           0     0.000         0     0.000
6           0     0.000         0     0.000
7           0     0.000         0     0.000
8           0     0.000         0     0.000
9           0     0.000         0     0.000
```

continued

```
                                     A TYPICAL KGIC RUN WITH ENGLISH DATA

                                    FREQUENCY OF OCCURRANCE OF EACH GRAPHEME

                                            RELATIVE FREQUENCY
GRAPHEME           2%        4%        6%        8%       10%       12%       14%       16%       18%     2
          ---------+---------+---------+---------+---------+---------+---------+---------+---------+-------

    A     ********************************************************
    B     *********
    C     **************
    D     **********************
    E     *****************************************************
    F     ************
    G     ************
    H     **************
    I     ***************************************
    K     ****
    L     ***************************************
    M     *****************
    N     *****************
    O     **************************
    P     **********************
    R     *******************************
    S     **************************
    T     *************************************************
    U     **************
    V     ****
    W     *******
    X     **
```

COST ESTIMATE

The computer costs for running the problem listed on the Sample Input were $3.30.

Charge to user = computer costs + network overhead
= $3.30 + network overhead

CONTENTS—KGIC

pages

1– 2 Identification & Abstract
3–10 User Instructions
11–16 I/O
17 Cost—Contents

DESCRIPTIVE TITLE	PL/I–FORMAC Interpreter
CALLING NAME	FORMAC
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	R. Tobey J. Baker R. Crews P. Marks K. Victor International Business Machines Corporation
LANGUAGE	PL/I–FORMAC
COMPUTER	IBM 360/67
PROGRAM AVAILABILITY	Source code available on tape
CONTACT	Dr. Daniel Bernitt, EIN Technical Representative, Computation Center, 105 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

FORMAC is a system for carrying out formal manipulations on mathematical expressions. This allows for the use of analytic as well as numeric techniques. The most important capability of FORMAC is its accommodation of mathematical expressions as symbolic entities at execution time. For example, the execution of the FORMAC program segment

```
LET( A = X + Y ** 2;
     B = 2;
     C = A/B + 2.8);
```

may be interpreted as assigning the alphanumeric value $(X+Y^2)/2 + 2.8$ to the FORMAC variable C.

FORMAC enables the user to analyze expressions by identifying coefficients, common denominators, lead operators, and some characteristics of the operands. Constants can be factored, left in a rational form, or converted to real notation. New expressions can be synthesized by the simplification, expansion and substitution of

continued

terms, as well as finding the derivatives of functions. The user can specify functions completely or partially in addition to making use of the PL/I functions. Procedures are available for transfering arguments between PL/I and FORMAC program segments.

REFERENCES

Tobey, R., Baker, J., Crews, R., Marks, P., Victor, K., "PL/I-FORMAC Interpreter User's Reference Manual," IBM Publ. 360D 03.3.004 (1967). Copies of the User's Manual will be available through The Pennsylvania State University Computation Center to any user with a PSU account number.

USER INSTRUCTIONS

The user should consult the reference listed.

FORMAC constants can be integer, floating-point, rational, or systems constants such as e, π, and i. Note that this latter constant enables FORMAC to handle imaginary expressions. FORMAC variables take on expressions as values that are retained as symbolic entities. Variables are either assigned or atomic—i.e., specified or unspecified. Expressions in FORMAC are written in essentially the same way as PL/I expressions but, whereas the latter evaluate to numbers, FORMAC expressions generally evaluate to new FORMAC expressions. There are four types of FORMAC functions: PL/I-like functions (such as SIN and COS), integer-valued functions (such as factorials and combinational expressions), user-defined functions, and function variables (unspecified functions). Functions with arguments that evaluate to constants can be either evaluated or retained as symbolic entities, at the user's option.

Program Deck

JOB-Control Language Statements and End-of-File Cards must begin in Col. 1; for FORMAC-PL/I statements, only Cols. 2—72 are used.

JOB Card will be punched by The Pennsylvania State University Computation Center personnel.

EXEC Card

Columns	*Contents*
1— 2	//
4— 7	EXEC
9—21	LIBRARY,PROG=
22—27	FORMAC

Data Card

Columns	*Contents*
1—17	//DATA.INPUT DD *

Program Deck

JOB Card
EXEC Card
Data Card
⋮
Program Deck
⋮
/*

continued

REFERENCES

Tobey, R., Baker, J., Crews, R., Marks, P., Victor, K., "PL/I-FORMAC Interpreter User's Reference Manual," IBM Publ. 360D 03.3.004 (1967). Copies of the **User's** Manual will be available through The Pennsylvania State University Computation Center to any user with a PSU account number.

SAMPLE INPUT

```
//AA112345 JOB 'S4002,R=600,T=60,FMAC','KNOBLE H'
// EXEC LIBRARY,PROG=FMACUT
//DATA.INPUT DD *
'OPTSET(LINELENGTH=SHORT)'
'COMMENT        IN THE FOLLOWING EXAMPLES "STRING=" INDICATES THE INPUT  '
'COMMENTSTRING:  THE UNDERLINED TWO-DIMENSIONAL EDITED OUTPUT IS THE'
'COMMENTRESULT OF PROCESSING BY THE INTERPRETATIVE FORMAC SYSTEM.'
'PAGE'
'COMMENT       GENERAL EXAMPLE.'
       'A = (X**2 + 7/3)**#P'
       'B = 3*SIN(A+1) + .3E-4*LOG(COS(Y))'
'COMMENT   UNSPECIFIED FUNCTIONS AND DERIVATIVES.'
       'A = F.(3*X**2) + LOG(X)'
       'B = DERIV(A,X,2)'
       'C = DERIV(ATANH(X**2) + X,X)'
       'D = DRV(H,(U,V,W),$(1),3, $(2),2, $(3),2)'
'COMMENT   TRANS OPTION FOR TRANCENDENTIAL FUNCTION EVALUATION.
'OPTSET(NOTRANS)'
       'A = 1/3'
       'B = SIN(X**4) + LOG(2*A) + SIN(#P/6)'
'OPTSET(TRANS)'
       'C = B'
'COMMENT   INT OPTION FOR COMBINATIONS, FACTORIAL, AND EVAL CAPABILITY.'
'OPTSET(NOINT)'
       'Y = COMB(N,K-1) + COMB(N,K)'
       'A = EVAL(Y,N,5,K,3)'
'OPTSET(INT)'
       'B = EVAL(Y,N,5,K,3)'
'COMMENT   FUNCTION DEFINITION AND EXPANSION APPLYING MULT. AND DIST. LAWS'
       'FNC(G) = $(1) + 2*SIN($(2))'
       'P = G(X**2+1,7)'
       'A = (X+4)**2 + 3*(2*X+7) * (7-3)'
       'B = EXPAND(A)'
'COMMENT   COMMON DENOM,. NUM, DENOM, LOW AND HIGH POWER, AND COEFF'
       'A = X/7 + 7 + X/W'
       'B = CODEM(A)'
       'Y = (7*X**5 + 7/3*X**3)/(SIN(X) + COS(X))'
       'A=NUM(Y)'  'B = DENOM(Y)'  'C = HIGHPOW(A,X)'  'D = LOWPOW(A,X)'
       'E = COEFF(A,X**5)'
'COMMENT   EDIT FEATURE.'
       'A = C1 + C2*X + C3*X**2'
'OPTSET(NOEDIT)'
       'C = A'
'COMMENT   INTEGER AND RATIONAL CONSTANTS MAY BE DEVELOPED OR SPECIFIED'
'COMMENTWITH AS MANY AS 2295 SIGNIFICANT DIGITS.'
       'Y = 11672713468795/5 * X**174'
       'A = DERIV(Y,X,5)'
       'B = EVAL(A,X,2+11/12)'
'COMMENT   Y=0: X<-1, X>1:  Y=1: -1.=X<0: Y=COS(X): 0<=X<1.'
       'Y = STEP(-1,X,0) + STEP(0,X,1) * COS(X)'
/*
```

SAMPLE OUTPUT

```
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNJBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNJBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA
HASP-II***HASP-II***HASP-II***...........START JOB 1817....AA112345......KNOBLE H.......................***HASP-II***HASP-II***  OR=AA

TIMF: 14:35:34  CATE: 01/21/70            H A S P   S Y S T E M   L O G


//AA112345 JOB 'S4002,R=60C,T=60,FMAC','KNOBLE H'


STRING='OPTSET(LINELENGTH=SHORT)';
LINFLENGTH=  64
      IN THE FOLLOWING EXAMPLES "STRING=" INDICATES THE INPUT
STRING;  THE UNDERLINED TWO-DIMENSIONAL EDITED OUTPUT IS THE
RESULT OF PROCESSING BY THE INTERPRETATIVE FORMAC SYSTEM.
```

continued

```
     GENERAL EXAMPLE.

STRING='A = (X**2 + 7/3)*#P';
             2
A = #P ( X    + 7/3 )
--------------------


STRING='B = 3*SIN(A+1) + .3E-4*LOG(COS(Y))';
                                              2
B = .3E-04 LN ( COS ( Y ) ) + 3 SIN ( #P ( X    + 7/3 ) + 1 )
-------------------------------------------------------------

   UNSPECIFIED FUNCTIONS AND DERIVATIVES.
            ↑abstract functions     f(x) is represented as F.(x)
STRING='A = F.(3*X**2) + LOG(X)';
              2
A = F.( 3 X     ) + LN ( X )
------------------------


STRING='B = DERIV(A,X,2)';
        (2)         2     2          2          (1)        2
B = 36 F     .( 3 X    ) X    - 1 / X    + 6 F     .( 3 X    )
--------------------------------------------------------------


STRING='C = DERIV(ATANH(X**2) + X,X)';
                   4
C = 2 X /  (  - X     + 1 ) + 1
---------------------------------


STRING='D = DRV(H.(U,V,W),$(1),3, $(2),2, $(3),2)';
      3  2  2
    (1  2  3  )
D = H              .( U, V, W )
--------------------------------

   TRANS OPTION FOR TRANCENDENTAL FUNCTION EVALUATION.

STRING='OPTSET(NOTRANS)';
OPTSET(NOTRANS)

STRING='A = 1/3';
A = 1/3
-------


STRING='B = SIN(X**4) + LOG(2*A) + SIN(#P/6)';
                                              4
B = LN ( 2/3 ) + SIN ( 1/6 #P ) + SIN ( X     )
-----------------------------------------------


STRING='OPTSET(TRANS)';
OPTSET(TRANS)

STRING='C = B';





               4
C = SIN ( X     ) + .09453489
-----------------------------

   INT OPTION FOR COMBINATIONS, FACTORIAL, AND EVAL CAPABILITY.

STRING='OPTSET(NOINT)';
OPTSET(NOINT)

STRING='Y = COMB(N,K-1) + COMB(N,K)';
Y = ( N; K ) + ( N; K - 1 )
---------------------------
```

continued

```
STRING='A = EVAL(Y,N,5,K,3)';
A = ( 5; 2 ) + ( 5; 3 )
-----------------------


STRING='OPTSET(INT)';
OPTSET(INT)

STRING='B = EVAL(Y,N,5,K,3)';
B = 20
------

   FUNCTION DEFINITION AND EXPANSION APPLYING MULT. AND DIST. LAWS

STRING='FNC(G) = $(1) + 2*SIN($(2))';
G = 2 SIN ( $(2) ) + $(1)
-------------------------


STRING='P = G(X**2+1,Z)';
     2
P = X   + 2 SIN ( Z ) + 1
-------------------------


STRING='A = (X+4)**2 + 3*(2*X+Z) * (Z-3)';
                                       2
A = 3 ( Z - 3 ) ( Z + 2 X ) + ( X + 4 )
-----------------------------------------


STRING='B = EXPAND(A)';
                               2     2
B =  - 9 Z - 10 X + 6 X Z + 3 Z   + X   + 16
--------------------------------------------

   COMMON DENOM., NUM, DENOM, LOW AND HIGH POWER  AND COEFF

STRING='A = X/Z + Z + X/W';
A = Z + X /  W + X /  Z
-----------------------


STRING='B = CODEM(A)';
B = ( X W + ( X + Z W ) Z ) / (  Z W )
--------------------------------------




STRING='Y = (Z*X**5 + 7/3*X**3)/(SIN(X) + COS(X))';
        5           3
Y =  ( X   Z + 7/3 X   ) / ( SIN ( X ) + COS ( X ) )
----------------------------------------------------


STRING='A=NUM(Y)';
     5           3
A = X   Z + 7/3 X
-------------------


STRING='B = DENOM(Y)';
B = SIN ( X ) + COS ( X )
-------------------------


STRING='C = HIGHPOW(A,X)';
C = 5
-----


STRING='D = LOWPOW(A,X)';
D = 3
-----
```

continued

```
STRING='E = COEFF(A,X**5)';
F = 7
-----

   EDIT FEATURE.

STRING='A = C1 + C2*X + C3*X**2';
                   2
A = C1 + X C2 + X   C3
---------------------


STRING='OPTSET(NOEDIT)';
OPTSET(NOEDIT)

STRING='C = A';
C = C1 + C2*X + C3*X**2
----------------------

   INTEGER AND RATIONAL CONSTANTS MAY BE DEVELOPED OR SPECIFIED
WITH AS MANY AS 2295 SIGNIFICANT DIGITS.

STRING='Y = 11672713468795/5 * X**174';
Y = X**174*2334542693759
-----------------------


STRING='A = DERIV(Y,X,5)';
A = X**169*3513748318138337313336720
-----------------------------------




STRING='B = EVAL(A,X,2+11/12)';
B = (720752901067543751748036225257161223586876421196893817372
-------------------------------------------------------------
953841030055920105250382345793678832323239245958570433346085570
-------------------------------------------------------------
507465119842170842320609136976317025501920713908186546167420740
-------------------------------------------------------------
7386691996264461274731368501587171417203988725358408990611693012
-------------------------------------------------------------
429128657458932138979434967041015625/557374165629320277805245157
-------------------------------------------------------------
2007421520389306011409217151508836170251045231030693760798803 9
-------------------------------------------------------------
27258048417240159298425788002423411242613443035327084411302454
-------------------------------------------------------------
10940816164117616172571033 6 )
----------------------------

   Y=0: X<-1, X>1;  Y=1: -1.=X<0;  Y=COS(X): 0<=X<1.

STRING='Y = STEP(-1,X,0) + STEP(0,X,1) * COS(X)';
Y = COS(X)*STEP(0,X,1) + STEP(-1,X,0)
------------------------------------


**NORMAL STOP WITH INPUT TO FMACUT EXHAUSTED**


HASP VERSION 2.M2C    51 CARDS READ    179 LINES PRINTED      0 CARDS PUNCHED      5 SECS NET CPU TIME
```

COST ESTIMATE

For the job listed on the Sample Output, the total running time on the central-processor unit was 10 seconds, at $0.11 per second. The chargeable computer time was $1.10.

Charge to user = computer time + network overhead
= $1.10 + network overhead

CONTENTS—FORMAC

pages
1– 2 Identification & Abstract
3– 4 User Instructions
5– 9 I/O
11 Cost—Contents

000 0050(a

DESCRIPTIVE TITLE	FORMAC Utility Program
CALLING NAME	FMACUT
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Robert Duquet Meteorology Department The Pennsylvania State University H.D. Knoble (assistance) The Pennsylvania State University Computation Center
LANGUAGE	PL/1—FORMAC
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Dr. Daniel L. Bernitt, EIN Technical Representative, Computation Center, 105 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

The FORMAC version available at The Pennsylvania State University is an interpretative system; i.e., FORMAC expressions are treated as character strings that are evaluated and executed at run time only. These character strings may be specified by reference to a PL/1 character-string variable. The PL/1 variable, in turn, may be constructed by execution of the PL/1 program in which the FORMAC statements are embedded. FMACUT merely capitalizes on the last-named feature. FORMAC statements are read by the preprocessed, precompiled, and prelink-edited program and are passed to the FORMAC package as character-string arguments. Note that this is made possible also because FORMAC variables need not (in fact *cannot*) be declared in the PL/1 program. Reference 1 should be consulted concerning more-detailed PL/1 FORMAC information.

FMACUT will execute FORMAC statements that are supplied as *input data*. It effectively divorces FORMAC from PL/1 insofar as the user is concerned. This has two advantages: (1) the user need have no knowledge of PL/1 to use FORMAC and (2) FORMAC is available without preprocessing, compilation, or link-editing (which

continued

000 0050(a)

saves approximately 50 seconds of run time for a null job on the IBM 360/67). The corresponding disadvantage is that PL/1 facilities and those (few) FORMAC statements involving a PL/1 direct interface are not available to the user.

REFERENCES

Tobey, R., Baker, J., Crews, R., Marks, P., and Victor, K., "PL/1 FORMAC Interpreter," IBM Publ. 360D 03.3.004 (1967).

USER INSTRUCTIONS

FMACUT may be executed via the catalogued procedure LIBRARY. The input is freeform; i.e., card boundaries are ignored. FORMAC statements are written within single quotes (') rather than within balanced parentheses following the word LET. For example, instead of using LET(X=A+B*COS(PL/2)), use 'X=A+B*COS(PL/2)'. Only one FORMAC statement may be enclosed within a given pair of single quotes. For example, instead of using LET(X+A**2+C*A—D; Y=COEFF (x,A**2)), use 'S=A**2+c*a—D' 'Y=COEFF(X,A**2)'. *NOTE:* The single quote terminating one FORMAC statement by one or more blanks and/or one comma. An individual FORMAC statement, exclusive of its enclosing quotes may be no longer than 800 characters in length, including blanks.

To facilitate setting FORMAC options, the OPTSET command may be used. OPTSET followed by the desired options (optionally enclosed within parentheses) must be enclosed within single quotes. For example, 'OPTSET INT EXPAND' and 'OPTSET(PROPER)' are acceptable. All options, including PRINT and NOPRINT, may be used and may be set, changed, and reset as often as desired. The default options are LINELENGTH=LONG (i.e., 120), PRINT, TRANS, INT, NOEXPND, EDIT, and IMPROPER. OPTSET is a reserved keyword.

To provide the capability of adding comments to the output, the command 'COMMENT-any comment' may be used. For example, 'COMMENT-PROOF BY INDUCTION' would result in this comment being printed. COMMENT is a reserved keyword and, like OPTSET, may not be used as a variable name.

To adjust the number of characters that will be printed out per line, two options are available: 'OPTSET(LINELENGTH=LONG)' will use a full page (120 characters per line) and 'OPTSET(LINELENGTH=SHORT)' will print out in a form suitable for 8½ X 11 in. paper (72 characters per line). When using the latter option, comments should be limited to 72 characters, including COMMENT.

Capabilities and Limitations

The commands SAVE and ATOMIZE are not available. In general, only those FORMAC statements that would normally be enclosed within parentheses following LET may be used. In particular, the features ARITH, INTEGER, IDENT, EXCH, and double quotes (") involving a direct PL/1 to FORMAC interface are not available. OPTSET and COMMENT are available as explained above. If a FORMAC expression is incorrectly structured with respect to syntax or logic, as described in the FORMAC manual,[1] the statement is bypassed, an error message is issued, and processing continues. One special character used by FORMAC—namely, the exclamation point (!)—is

continued

not available on the QN (normal) print train. Therefore, this character, used as a factorial sign when the INT option is *off*, will print as a blank. This limitation, however, is not probelm when using a terminal via RJE.

The transcendental-function names that are available are the following: SIN, COS, SINH, COSH, ATAN, ATANH, ATAN(x,y), LOG, LOG10, LOG2, ERF, EXP, SQRT, SIND, COSD, TAN, TAND, TANH, ATAND, ATAND(x,y), and ERFC. In addition, as part of the functional expression, the constants #P=Pi, #E=e (base of natural logs), and #i=$\sqrt{-1}$ will be recognized. [E.g., FORMAC will transform SIN(#P/6) before computation to 1/2.] The functions FAC(n) = n! and $\text{COMB}(n,n_1,n_2,\cdots,n_k) = [n(n-1)\cdots(n-n_1-\cdots-n_k+1)]/[n!n_1!\cdots n_k!]$ also are available for n and k non-negative integers. The absolute-value function ABS(X) is *not* available; however, this operation may be effected by SQRT(X**2).

The number of significant digits in any floating-point constants may not exceed eight. The number of variables and degree of expansion are limited only by storage. Most of these limitations can be eliminated for a particular case by modifying FMACUT.

Program deck and Control Cards are the same for FMACUT and FORMAC, except that Cols. 22–27 of the EXEC Card contain FMACUT.

REFERENCES

1. Tobey, R., Baker, J., Crews, R., Marks, P., and Victor, K., "PL/1 FORMAC Interpreter," IBM Publ. 360D 03.3.004 (1967).

SAMPLE INPUT

```
// EXEC LIBRARY,PROG=FMACUT
//DATA.INPUT DD *
'OPTSET(LINELENGTH=SHORT)'
'COMMENT       IN THE FOLLOWING EXAMPLES "STRING=" INDICATES THE INPUT  '
'COMMENTSTRING;  THE UNDERLINED TWO-DIMENSIONAL EDITED OUTPUT IS THE'
'COMMENTRESULT OF PROCESSING BY THE INTERPRETATIVE FORMAC SYSTEM.'
'PAGE'
'COMMENT      GENERAL EXAMPLE.'
      'A = (X**2 + 7/3)*#P'
      'B = 3*SIN(A+1) + .3E-4*LOG(COS(Y))'
'COMMENT   UNSPECIFIED FUNCTIONS AND DERIVATIVES.'
      'A = F.(3*X**2) + LOG(X)'
      'B = DERIV(A,X,2)'
      'C = DERIV(ATANH(X**2) + X,X)'
      'D = DRV(H.(U,V,W),$(1),3, $(2),2, $(3),2)'
'COMMENT   TRANS OPTION FOR TRANCENDENTAL FUNCTION EVALUATION.'
'OPTSET(NOTRANS)'
      'A = 1/3'
      'B = SIN(X**4) + LOG(2*A) + SIN(#P/6)'
'OPTSET(TRANS)'
      'C = B'
'COMMENT   INT OPTION FOR COMBINATIONS, FACTORIAL, AND EVAL CAPABILITY.'
'OPTSET(NOINT)'
      'Y = COMB(N,K-1) + COMB(N,K)'
      'A = EVAL(Y,N,5,K,3)'
'OPTSET(INT)'
      'B = EVAL(Y,N,5,K,3)'
'COMMENT   FUNCTION DEFINITION AND EXPANSION APPLYING MULT. AND DIST. LAWS'
      'FNC(G) = $(1) + 2*SIN($(2))'
      'P = G(X**2+1,Z)'
      'A = (X+4)**2 + 3*(2*X+Z) * (Z-3)'
      'B = EXPAND(A)'
'COMMENT   COMMON DENOM., NUM, DENOM, LOW AND HIGH POWER, AND COEFF'
      'A = X/Z + Z + X/W'
      'B = CODEM(A)'
      'Y = (Z*X**5 + 7/3*X**3)/(SIN(X) + COS(X))'
      'A=NUM(Y)'  'B = DENOM(Y)'  'C = HIGHPOW(A,X)'  'D = LOWPOW(A,X)'
      'E = COEFF(A,X**5)'
'COMMENT   EDIT FEATURE.'
      'A = C1 + C2*X + C3*X**2'
'OPTSET(NOEDIT)'
      'C = A'
'COMMENT   INTEGER AND RATIONAL CONSTANTS MAY BE DEVELOPED OR SPECIFIED'
'COMMENTWITH AS MANY AS 2295 SIGNIFICANT DIGITS.'
      'Y = 11672713468795/5 * X**174'
      'A = DERIV(Y,X,5)'
      'B = EVAL(A,X,2+11/12)'
'COMMENT   Y=0: X<-1, X>1;  Y=1: -1.=X<0; Y=COS(X): 0<=X<1.'
      'Y = STEP(-1,X,0) + STEP(0,X,1) * COS(X)'
/*
```

SAMPLE OUTPUT

```
STRING='OPTSET(LINELENGTH=SHORT)';
LINFLENGTH=  64
       IN THE FOLLOWING EXAMPLES "STRING=" INDICATES THE INPUT
STRING:  THE UNDERLINED TWO-DIMENSIONAL EDITED OUTPUT IS THE
RESULT OF PROCESSING BY THE INTERPRETATIVE FORMAC SYSTEM.


      GENERAL EXAMPLE.

STRING='A = (X**2 + 7/3)*#P';
          2
A = #P ( X   + 7/3 )
--------------------


STRING='B = 3*SIN(A+1) + .3E-4*LOG(COS(Y))';
                                            2
B = .3E-04 LN ( COS ( Y ) ) + 3 SIN ( #P ( X   + 7/3 ) + 1 )
--------------------------------------------------------------

   UNSPECIFIED FUNCTIONS AND DERIVATIVES.

STRING='A = F.(3*X**2) + LOG(X)';
          2
A = F.( 3 X   ) + LN ( X )
---------------------------


STRING='B = DERIV(A,X,2)':
        (2)         2    2         2      (1)         2
B = 36 F      .( 3 X  ) X   - 1 / X  + 6 F      .( 3 X   )
----------------------------------------------------------


STRING='C = DERIV(ATANH(X**2) + X,X)';
                 4
C = 2 X /  (  - X   + 1 ) + 1
------------------------------


STRING='D = DRV(H.(U,V,W),$(1),3, $(2),2, $(3),2)';
        3  2  2
     (1  2  3  )
D = H            .( U, V, W )
------------------------------

   TRANS OPTION FOR TRANCENDENTAL FUNCTION EVALUATION.

STRING='OPTSET(NOTRANS)';
OPTSET(NOTRANS)

STRING='A = 1/3';
A = 1/3
-------


STRING='B = SIN(X**4) + LOG(2*A) + SIN(#P/6) ;
                                         4
B = LN ( 2/3 ) + SIN ( 1/6 #P ) + SIN ( X   )
----------------------------------------------


STRING='OPTSET(TRANS)';
OPTSET(TRANS)

STRING='C = B';
```

continued

```
          4
C = SIN ( X    ) + .09453489
----------------------------

   INT OPTION FOR COMBINATIONS, FACTORIAL, AND EVAL CAPABILITY.

STRING='OPTSET(NOINT)';
OPTSET(NOINT)

STRING='Y = COMB(N,K-1) + COMB(N,K)';
Y = ( N; K ) + ( N; K - 1 )
----------------------------


STRING='A = EVAL(Y,N,5,K,3)';
A = ( 5; 2 ) + ( 5; 3 )
------------------------


STRING='OPTSET(INT)';
OPTSET(INT)

STRING='B = EVAL(Y,N,5,K,3)';
B = 20
------

   FUNCTION DEFINITION AND EXPANSION APPLYING MULT. AND DIST. LAWS

STRING='FNC(G) = $(1) + 2*SIN($(2))';
G = 2 SIN ( $(2) ) + $(1)
--------------------------


STRING='P = G(X**2+1,Z)';
    2
P = X   + 2 SIN ( Z ) + 1
--------------------------


STRING='A = (X+4)**2 + 3*(2*X+Z) * (Z-3)';
                                       2
A = 3 ( Z - 3 ) ( Z + 2 X ) + ( X + 4 )
-----------------------------------------


STRING='B = EXPAND(A)';
                              2     2
B =  - 9 Z - 10 X + 6 X Z + 3 Z   + X   + 16
---------------------------------------------

   COMMON DENOM., NUM, DENOM, LOW AND HIGH POWER, AND COEFF

STRING='A = X/Z + Z + X/W';
A = Z + X /  W + X /  Z
-----------------------


STRING='B = CODEM(A)';
B = ( X W + ( X + Z W ) Z ) / (  Z W )
--------------------------------------
```

continued

```
STRING='Y = (Z*X**5 + 7/3*X**3)/(SIN(X) + COS(X))';
        5             3
Y =   ( X    Z + 7/3 X    ) / ( SIN ( X ) + COS ( X ) )
-------------------------------------------------------


STRING='A=NUM(Y)';
     5            3
A = X    Z + 7/3 X
-------------------


STRING='B = DENOM(Y)';
B = SIN ( X ) + COS ( X )
-------------------------


STRING='C = HIGHPOW(A,X)';
C = 5
-----


STRING='D = LOWPOW(A,X)';
D = 3
-----


STRING='E = COEFF(A,X**5)';
E = Z
-----

   EDIT FEATURE.

STRING='A = C1 + C2*X + C3*X**2';
                        2
A = C1 + X C2 + X    C3
----------------------


STRING='OPTSET(NOEDIT)';
OPTSET(NOEDIT)

STRING='C = A';
C = C1 + C2*X + C3*X**2
-----------------------

   INTEGER AND RATIONAL CONSTANTS MAY BE DEVELOPED OR SPECIFIED
WITH AS MANY AS 2295 SIGNIFICANT DIGITS.

STRING='Y = 11672713468795/5 * X**174';
Y = X**174*2334542693759
------------------------


STRING='A = DERIV(Y,X,5)';
A = X**169*351374831813833791336720
-----------------------------------
```

continued

```
STRING='B = EVAL(A,X,2+11/12)';
B = (720752901067543751748096225257161223588876421196893817372
------------------------------------------------------------
953841030055920105250382345753678832323239245958570433346085570
------------------------------------------------------------
507465119842170842320609136976317025501920713908186546157420740
------------------------------------------------------------
738669198626461274731368501587171417203988725358408990611693012
------------------------------------------------------------
429128657458932138979434567041015625/5573741656293202778052451 57
------------------------------------------------------------
200742152038930601140921715150883619702510452310306937607988039
------------------------------------------------------------
272580484172401592984257880024234112442613443035327084411302454
------------------------------------------------------------
10940816164117616172571 0336 )
------------------------------

   Y=0: X<-1, X>1;  Y=1: -1.=X<0; Y=COS(X): 0<=X<1.

STRING='Y = STEP(-1,X,0) + STEP(0,X,1) * COS(X)';
Y = COS(X)*STEP(0,X,1) + STEP(-1,X,0)
-------------------------------------


**NORMAL STOP WITH INPUT TO FMACUT EXHAUSTED**



HASP VERSION 2.M28    52 CARDS READ    236 LINES PRINTED      0 CARDS PUNCHED      5 SECS NET CPU TIME
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central-processor unit was 10 seconds, at $0.11 per second. The chargeable computer time was $1.10.

Charge to user = computer time + network overhead
= $1.10 + network overhead

CONTENTS—FMACUT

pages

1– 2	Identification & Abstract
3– 4	User Instructions
5– 9	I/O
11	Cost—Contents

DESCRIPTIVE TITLE	Collection of Statistical Routines
CALLING NAME	STPAC
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	William H. Verity Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

STPAC consists logically of two sections: a monitor and a collection of statistical routines. The monitor is a program that will read all Control Cards, save the original data if required or requested, save any intermediate results that may be needed later on, and then transfer control to the first routine requested. When this routine is finished, it returns control to the monitor, which then looks to see if there is more work to be done. This work can be of several different types, as follows.

1. Using the same data deck (but possibly different columns of it), execute the same routine again.
2. Using the same data deck, execute another routine. Thus, one deck can be subjected to several types of analyses during one run.
3. Execute another program, possibly the same one as before, on a new set of data.
4. Using the results of a previous routine, continue to another program that takes these results as input; e.g., output from correlation can be the input to regression or factor analysis.

The statistical routines available include a variety for summary

continued

86

of data, frequency analysis, tests of significance, analyses of variance, and rank and product-moment correlation coefficients. See individual abstracts for EIN Nos. 000 0051(a)-(s) for the details of the individual statistical routines.

REFERENCES

See individual abstracts for EIN numbers 000 0051 (a)-(s).

STPAC ROUTINES

EIN No.	*Calling Name*
000 0051(a)	FAWCS
(b)	STSUM
(c)	TTEST
(d)	PPMCR
(e)	SIGPP
(f)	PARCOR
(g)	CANON
(h)	UPREG/DNREG
(i)	FANAL
(j)	VARMX
(k)	PHICO
(l)	KETAU
(m)	SPRHO
(n)	MNWHT
(o)	KRWAL
(p)	ANOVUM
(q)	ANOVES
(r)	AOVRM
(s)	BARTL

USER INSTRUCTIONS

To use any of the programs of the statistical package, *STPAC MUST BE SPECIFIED AS THE LIBRARY NAME.* Programs available as part of this package are identified and described in separate writeups [EIN Nos. 000 0051(a)–(s)].

Control Cards and Their Functions

Problem information is given through the use of Control Cards. Each Control Card transmits one type of information and conveys to the user its function. The information on a Control Card is contained in one or more fields, where a field is defined to be one or more contiguous nonblank characters. Fields are separated from one another by one or more blank spaces or commas, thus allowing the user to punch the Control Cards without regard to specific columns, although the fields must be in the specified order. The only restriction is that the first field must begin in Col. 1. If any field contains special characters or blanks, it must be enclosed in single quotation marks.

To execute any given analysis, two types of Control Cards will be needed: those applicable to all of the programs in the package and those specific to the program at hand. The Control Cards that can apply to *all* programs are described next. The Control Cards that are peculiar to *an individual* program are described in the appropriate program writeup. Information printed in capital letters must appear on the Control Card letter for letter as it is typed and all fields must be in the order indicated.

Execute Card (required for each problem)

Columns	*Contents*
1– 7	EXECUTE
9–13	Name of program to be executed

NOTE: This must be the first card for *each* problem under STPAC.

Comments Card(s) (optional)

Columns	*Contents*
1	*
2–80	Information to be printed on a page immediately preceding output (for documentation)

Any number (or none) of these cards may be used.

continued

Title Card (optional)

Columns	*Contents*
1– 5	TITLE
6–80	Heading to be printed on each section of output

Variable Card (may or may not be required)

Columns	*Contents*
1– 9	VARIABLES
11–12	Number of variables for the problem
14–22	SEQUENCED
24–25	Number of variables on each Data Card (≤ number in Cols. 11–12)
27–32	PERCARD

NOTE: If Cols. 14–32 of this card are specified, then it is inferred that case and sequence numbers (strictly increasing only) are on each Data Card, each case number being identified in the first card field and the sequence number within the case being given in the second card field. If the case and sequence numbers are missing (or it is not desired that they be checked), entries in these columns *must* be omitted. In this instance, either all variables fit on one card or the Format Card (see below) controls the reading of more than one card.

Name Card(s) (optional; first form)

Columns	*Contents*
1–80	VARIABLE NAMES i=xxx,j=yyy, etc. or TREATMENT NAMES i=xxx,j=yyy, etc.

One or more of these cards may be used to assign alphanumeric names to each variable and/or treatment. Variable (treatment) i will be called xxx, j will be called yyy, etc., where xxx and yyy represent some combinations of eight or fewer characters, *at least one of which must be nonnumeral.* If the card is omitted, variables (treatments) will be numbered sequentially. It is also possible to name *some* variables (treatments) and to allow the remaining ones to be numbered. In addition, xxx, yyy, etc., may be labelled with a two-word term if it totals no more than eight characters and has single quotation marks around it.

continued

Examples (starting in Col. 1)

```
 VARIABLE NAMES 1=AGE,2=PARTY
TREATMENT NAMES 1=HEIGHT,2='HAIR COL'
```

Name Card(s) (optional; second form)

Columns	*Contents*
1–80	VARIABLE NAMES TRANSFER or TREATMENT NAMES TRANSFER

This card indicates that the names from the immediately preceding problem are to be used in the present problem (especially useful when results from one problem are data for another).

Format Card(s) (may or may not be required)

Columns	*Contents*
1– 6	FORMAT
8	Number of cards used in format specification (≤ 3)
10–80	Beginning of format specification (any valid FORTRAN FORMAT)

NOTE: If the number of cards required is one, the format specification may begin in Col. 8 (and the 1 need not be specified). If more than one card is required, the word FORMAT need not be repunched, thus making all 80 columns of the second (and third) card available for use.

Begin-Data Card (required if Data Cards are to be used; must be omitted if data from previous problem are to be used)

Columns	*Contents*
1–10	BEGIN DATA

This card must immediately precede the data deck. Previous data are no longer available after this card has been read.

End-Data Card (required if Data Cards are being used)

Columns	*Contents*
1– 8	END DATA

This card must immediately follow the data deck.

continued

Same-Data Card (optional)

Columns	*Contents*
1– 9	SAME DATA

This card indicates that the *last* set of raw data is to be used, allowing a data deck to be used over and over again. When this card is used, there *cannot* be either a Begin- or End-Data Card.

Input Card (optional)

Columns	*Contents*
1–10	INPUT FROM
12–16	Name of program already run (during the *present* run) from which the input to this program is coming (used when the results of one program are the input to another). There may be cases in which the input may be from *both* another program and card input. Then, both the INPUT FROM *and* BEGIN DATA—END DATA sequence would be used.

Stop Card (required once for each submission to the package)

Columns	*Contents*
1– 4	STOP (indicating the end of input to STPAC)

Execution of all programs will terminate after this card is read.

Job Deck Structure

```
EXECUTE xxxxx (required)
Comments Card(s)―――――――――――――――――――┐            ――――――――――――┐
Title Card                           |                        |
Variable Card                        |                        |  Repeat
Name Card(s)                         |                        |  for as
Format Card(s)                       |  May be in             |  many
Other cards (required or optional    |  any order             |  problems
           for program xxxxx)        |                        |  (programs)
Same-Data Card                       |                        |  as neces-
Input Card ――――――――――――――――――――――――――┘                        |  sary
Begin-Data Card ┐                                             |
Data Card(s)    |  Not used if Same-Data Card is used or      |
End-Data Card___|  Input Card specifies all required input ―――┘
Stop Card (required at end of STPAC input)
```

NOTE: For each problem, there *must* be either a Same-Data Card, an Input Card, or a deck of Data Cards (beginning with the Begin-Data Card and ending with the End-Data Card).

SAMPLE INPUT/OUTPUT

Sample Input and Output are specific to the various routines. See items 000 0051(a)-(s).

COST ESTIMATE

Inappropriate for STPAC alone; see individual routines EIN Nos. 000 0051 (a)–(s)

CONTENTS—STPAC

pages

1–2 Identification & Abstract
3–5 User Instructions
7 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Frequency Analysis with Chi Square
CALLING NAME	(STPAC) FAWCS
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	C.B. Broderick Department of Family Relations, The Pennsylvania State University William H. Verity The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

FAWCS computes (1) discrete, two-dimensional frequency-distribution tables (often called *contingency tables*) whose cells contain an actual frequency and, optionally, (2) expected frequencies and/or percentages. There is no missing-data code inherent to the program, but the user may specify that certain categories of his variables be ignored or combined with one another. If the user has any 2X2 tables, he may request that the Yates correction factor be used in computing his chi-square value.

FAWCS will crosstabulate one variable, the *row* variable, with variables called *column* variables. It can work with up to 50 total variables for any one problem, separately crosstabulating the row variable with each of the column variables. The row variable can have up to 31 categories where 0 is a valid category. Thus, the row variable can be coded 0 through 30. Each column variable can have a maximum of 13 categories where, again, 0 is a valid code. Therefore, column variables can be coded 0 through 12. *Any data outside of these ranges will be ignored.*

continued

REFERENCES

Edwards, A.L., *Experimental Design in Psychological Research* (Holt Rinehart and Winston, New York, 1962).

Snedecor, G.W., *Statistical Methods* (Iowa State University Press, Ames, Ia., 1961).

Wert, J.E., Neidt, C.O., and Ahmann, J.S., *Statistical Methods in Educational and Psychological Research* (Appleton—Century—Crofts, New York, 1954).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

In addition to the Control Cards applicable to all programs in STPAC, FAWCS uses the Control Cards that are described in the following. There is no standard format for raw data, but if totals data are used they must be punched as described under the PUNCH option. *A Variables Card is required for FAWCS.*

Row-Variable Card (required)

Columns	*Contents*
1–12	ROW VARIABLE
14–80	Variable to be used as *row* variable. All others will become column variables. It may be specified by either the name or the number assigned to it. (If Name Cards were not used. the reference to the variable *by name* would be permitted.)

Percentages Card (optional)

Columns	*Contents*	
1–11	PERCENTAGES	
13–15	ROW	one of these
	or	
13–18	COLUMN	

This card indicates that percentages are to be computed on the basis of row (column) totals. Only one choice may be made for any one problem.

Chi-Square Card (optional)

Columns	*Contents*
1–10	CHI-SQUARE

This card indicates that a chi-square value is to be printed for each table, as well as the probability of exceeding this value by chance.

continued

Yates-Correction-Factor Card (optional)

Columns	*Contents*
1– 5	YATES

This card indicates that the Yates correction factor should be used in any 2X2 tables that are developed. It will be used *only* in computing the chi square.

Expected-Frequency Card (optional)

Columns	*Contents*
1– 8	EXPECTED

This card causes the expected cell frequencies to be printed.

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card causes the raw cell frequencies to be punched onto cards so that it is possible to go back and either combine or ignore different or new combinations of row and column categories without rereading the raw data. In order that a minimum number of cards have to be repunched by hand, the following cards are punched.

```
EXECUTE FAWCS
VARIABLES 2
VARIABLE NAMES 1=xxxxx,2=wwwww
ROW VARIABLE xxxxx
TOTALS (see below)
BEGIN DATA  |
   :        |  actual totals from the raw table
END DATA ___|
```

Thus, all that must be added is a Title Card. After adding or ignoring any other cards (including optional Control Cards), the deck is ready for resubmission. Totals are punched according to a format of 20I4, as follows.

Columns	*Contents*
1– 4	Row number
6– 9	Column number
13	1
14–17	Total for this row–column combination

continued

NOTE: The PUNCH option is recommended only for exceptionally large amounts of data. Otherwise, it is easier simply to resubmit the raw data.

Totals Card (optional)

Columns	*Contents*
1– 6	TOTALS

This card indicates that totals punched in a previous run of FAWCS (or otherwise) are to be used. Totals must be punched in the 20I4 format described above.

Ignore-Categories Card (optional; one or more as required—first form)

Columns	*Contents*
1–80	IGNORE xxxxx CATEGORIES i j k ···

This card causes categories i,j,k,··· of the variable xxxxx to be ignored, where xxxxx is either the name or the number of a variable. A preliminary table with *all* categories will be printed, followed by a final table in which the specified categories will not appear. All chi squares, percentages, and expected frequencies will be based on the final table.

Ignore-Categories Card (optional; second form)

Columns	*Contents*
1–80	IGNORE $COLUMN$ CATEGORIES i j k ...

This card indicates that categories i,j,k,··· of *all column variables* are to be ignored. To ignore row-variable categories, the *first form* of this card must be used.

Combine-Categories Card (optional; one or more as required—first form)

Columns	*Contents*
1–80	COMBINE xxxxx CATEGORIES i j k ···

This card causes categories i,j,k,··· of variable xxxxx to be combined to form a new category, where xxxxx is xxxxx is either the name or the number of a variable. The new category will carry a number that is the smallest of those combined.

continued

Combine-Categories Card (optional; second form)

Columns	*Contents*
1–80	COMBINE $COLUMN$ CATEGORIES i j k ···

This form indicates that, for each column variable, categories i,j,k,··· are to be combined to form a new category. Again, to combine row-variable categories, the *first form* of this card must be used.

NOTE: If there is an inconsistency in the Combine- and Ignore-Categories Cards, only the preliminary table will be printed and cards will be punched.

Format Card (required for raw data; not permitted for totals)

The form of this card is as described in the general discussion of Control Cards for STPAC. For FAWCS, the format can specify only integer data (I notation). If case and sequence numbers are required, the format must allow for them.

Description of Output

The printed output for a FAWCS problem is a series of two-dimensional tables, one table for each column variable. Each table is headed by the title that the user has supplied. The categories of each column variable are listed across the top of the table, along with the name of the column variable. The categories of the row variable are listed down the lefthand margin of the table, along with the name of the row variable. The body of the table contains the observed frequencies for each cell and percentages and expected frequencies if requested. Below each table, if requested, the chi-square value is printed, along with its degrees of freedom and the probability of exceeding this value by chance.

If any Combine- or Ignore-Categories Cards were present, a preliminary table is printed. It contains the same title and headings, but only the observed frequencies appear in the body of the table. At the bottom of this table are listed the categories that are to be ignored and combined. Then, a final table in which the combining and ignoring has taken place is printed on a new page. All calculations are based on this final table.

Card output is produced where applicable, as described under the Punch Card.

SAMPLE INPUT

```
EXECUTE FAMCS
VARIABLES 2
FORMAT(4X,2I4)
VARIABLE NAMES 1=COLLEGE,2=RESIDENCE
TITLE SAMPLE PROBLEM FOR STREC PROGRAM FAMCS
ROW VARIABLE COLLEGE
CHISQUARE
BEGIN DATA
   1   4   1 14117.
   2   4   2 14220.
   3   4   1  5637
   4   2   1  7736.
   5   2   2 11875.
   6   1   0  8137.
   7   4   1 12175.
   8   2   2  5550.
   9   4   1 14441.
  10   1   0  7875.
  11   5   1  8310.
  12   3   1  7928.
  13   1   2 14454.
  14   1   1 14701.
  15   2   1  9498.
  16   3   1  5656.
  17   1   1 11275.
  18   4   2  5436.
  19   1   2  8429.
  20   3   2  5637.
  21   5   1  8826.
  22   2   2 13138.
  23   5   1  7066.
  24   4   1 13917.
  25   5   2 10570.
  26   1   2  5410.
  27   1   1 11101.
  28   3   1  8560.
  29   5   1  9266.
  30   1   2  9863.
  31   2   1  8929.
  32   1   2  7051.
  33   3   0 13634.
  34   5   2  6100.
  35   1   2 10494.
  36   1   1  9692.
  37   0   0 13194.
  38   3   1 12369.
  39   1   2 14916.
  40   1   1  8462.
  41   3   1 11940.
  42   2   1 14822.
  43   2   2  8737.
  44   5   1  6468.
  45   2   1 13314.
  46   4   2 10950.
  47   5   1  5437.
  48   1   1 11110.
  49   2   0  6678.
  50   1   1 11625.
END DATA
EXECUTE FAMCS


SAME DATA
FORMAT(4X,2I4)
TITLE ILLUSTRATE SAME DATA OPTION
CHISQUARE
PERCENTAGES ROW
IGNORE COLLEGE CATEGORIES 0
IGNORE $COLUMNS CATEGORIES 0
VARIABLES 2
COMBINE COLLEGE CATEGORIES 1 2
COMBINE 1 CATEGORIES 3 4 5
YATES
STOP
```

FILMED FROM BEST AVAILABLE COPY

100

SAMPLE OUTPUT

```
                                                      FREQUENCY DISTRIBUTION TABLE

                                                           WITH CHI-SQUARE

                                                  SAMPLE PROBLEM FOR STPAC PROGRAM FAWCS

          EACH CELL OF THE TABLE CONTAINS AN ACTUAL FREQUENCY ON THE FIRST LINE

  *  CATEGORIES*
   *     OF    *
    *  COLUMN  *
     *VARIABLE *
      *RESIDENC*
       *   2   *
        *      *
         *     *
CATEGORIES*    *
   OF     *    *
  ROW      *   *                        ROW
 VARIABLE   *  *    0        1        2       TOTAL
  1  COLLEGE    **
***************************************************
               *
     0         *    1        0        0        1
               *
     1         *    2        7        7       16
               *
     2         *    1        5        4       10
               *
     3         *    1        5        1        7
               *
     4         *    0        5        3        8
               *
     5         *    0        6        2        8
               *
   COLUMN      *
    TOTAL      *    5       28       17       50
THE VALUE OF CHI-SQUARE FOR   10 DEGREES OF FREEDOM IS EQUAL TO   13.768

THE PROBABILITY OF EXCEEDING THIS VALUE OF CHI-SQUARE BY CHANCE IS 0.184
```

continued

```
                                                         FREQUENCY DISTRIBUTION TABLE

                                                         ILLUSTRATE SAME DATA OPTION

                                                               PRELIMINARY TABLE

   *  CATEGORIES*
    *     OF    *
     *  COLUMN  *
      *VARIABLE *
       *RESIDENC*
        *   2   *
         *      *
          *     *
CATEGORIES*     *
   OF      *    *
  ROW       *   *                        ROW
VARIABLE     *  *    0        1        2    TOTAL
 1  COLLEGE    **
*****************************************************
                *
     0          *    1        0        0       1
                *
     1          *    2        7        7      16
                *
     2          *    1        5        4      10
                *
     3          *    1        5        1       7
                *
     4          *    0        5        3       8
                *
     5          *    0        6        2       8
                *
  COLUMN        *
   TOTAL        *    5       28       17      50

THE FOLLOWING   ROW  CATEGORIES WERE IGNORED -  C,

THE FOLLOWING COLUMN CATEGORIES WERE IGNORED -  C,

THE FOLLOWING   ROW  CATEGORIES WERE COMBINED TO FORM CATEGORY  1 -  1,  2,

THE FOLLOWING   ROW  CATEGORIES WERE COMBINED TO FORM CATEGORY  3 -  3,  4,  5,
```

continued

```
                         PERCENT-FREQUENCY DISTRIBUTION TABLE

                                  WITH CHI-SQUARE

                              ILLUSTRATE SAME DATA OPTION

          EACH CELL OF THE TABLE CONTAINS A PERCENTAGE FREQUENCY ON THE FIRST LINE AND AN ACTUAL FREQUENCY ON THE SECOND LINE
          PERCENTAGES ARE BASED ON   ROW  TOTALS.

  *  CATEGORIES*
   *     OF    *
    *  COLUMN  *
     *VARIABLE *
      *RESIDENC*
       *   2   *
        *      *
         *     *
CATEGORIES*    *
   OF     *    *
  ROW      *   *                 ROW
 VARIABLE   *  *   1       2     TOTAL
 1  COLLEGE   **
*******************************************
               *
      :        * 52.2    47.8    51.1
               *   12      11      23
               *
      3        * 72.7    27.3    48.9
               *   16       6      22
               *
  COLUMN       * 62.2    37.8
   TOTAL       *   28      17      45

THE VALUE OF CHI-SQUARE FOR    1 DEGREES OF FREEDOM IS EQUAL TO    1.241

YATES CORRECTION FACTOR WAS USED IN OBTAINING THIS VALUE OF CHI SQUARE

THE PROBABILITY OF EXCEEDING THIS VALUE OF CHI-SQUARE BY CHANCE IS 0.265
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 5 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.55 + network overhead

CONTENTS—(STPAC) FAWCS

pages	
1– 2	Identification & Abstract
3– 6	User Instructions
7–10	I/O
11	Cost—Contents

DESCRIPTIVE TITLE	Statistical Summary
CALLING NAME	(STPAC) STSUM
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

STSUM computes, in double-precision arithmetic, the sample size, total, mean, standard deviation, population-variance estimate, standard error, sum of squares, coefficient of variation, third and fourth moments about the origin, third and fourth central moments, alpha (square root of beta 1), alpha 4 (beta 2), momental skewness, and kurtosis. Missing data will be permitted.

Equations

$$\left.\begin{aligned} V_1 &= \Sigma X/N \\ V_2 &= \Sigma X^2/N \\ V_3 &= \Sigma X^3/N \\ V_4 &= \Sigma X^4/N \end{aligned}\right] \quad \begin{array}{ll} \text{moments about origin} & \\ V_1 = \bar{X} & \text{mean} \\ \Sigma X & \text{total} \end{array}$$

$$\left.\begin{aligned} \mu_2 &= V_2 - V_1^2 \\ \mu_3 &= V_3 - 3V_1V_2 + 2V_1^3 \\ \mu_4 &= V_4 - 4V_1V_3 + 6V_1^2V_2 - 3V_1^4 \\ \alpha_3 &= \sqrt{\beta_1} = (\mu_3^2/\mu_2^3)^{1/2} \\ \alpha_4 &= \beta_2 = \mu_4/\mu_2 \end{aligned}\right] \quad \text{moments about mean}$$

$\alpha_3/2$ momental skewness

$(\alpha_4 - 3)/2$ kurtosis

$\Sigma X^2 - (\Sigma X)^2/N$ sum of squares

continued

REFERENCES

Beyer, W.H., Ed., *Handbook of Tables for Probability and Statistics* (Chemical Rubber Co., Cleveland, 1966), p.3.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential to STSUM. Begin- and End-Data Cards also are necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional.

Missing-Data Card (optional)

Columns	*Contents*
1–12	MISSING DATA
14–21	x, any number, up to eight digits including a decimal point, that will be read as a code for missing data.

Include this card only if some data are missing. If this card is omitted, it will be assumed that all fields contain data (blanks will be read as zero). If this card is included and Cols. 14–21 are blank, zeros and blanks in the Data Cards will be treated as missing data. If Cols. 14–21 are punched, the number x in a data field will be treated as a case of missing data.

Example (starting in Col. 1)

```
MISSING DATA 99.9
```

This would provide for a case where Data Cards can contain instances of zero observations and instances of missing data. All data fields previously blank would then be coded 99.9 on the Data Cards.

Description of Input Data

Data must be punched according to the format given on the Format Card. A case and sequence number should be punched on each card (but only if required, as described in the STPAC writeup), as well as the data points in F or E format. The number of a variable is determined by the order in which it is read, which is determined by the format supplied.

continued

Description of Output

Statistics are printed in two tables. All variables are numbered and also are named according to the Variable-Name(s) Card, if it is included.

REFERENCES

Beyer, W.H., Ed., *Handbook of Tables for Probability and Statistics* (Chemical Rubber Co., Cleveland, 1966), p.3.

SAMPLE INPUT

```
EXECUTE STSUM
TITLE TEST ONE
VARIABLES 4 SEQUENCED 2 PERCARD
FORMAT (2I1.2F1.0)
BEGIN DATA
4256
4437
1138
1654
END DATA
```

SAMPLE OUTPUT

```
                                                     TEST ONE

VARIABLE    NUMBER      TOTAL          MEAN          ST.DEV.   VARIANCE ESTIMATE   ST.ERROR     SUM OF SQUARES     CV. PCT.
       1         2   0.800000D 01   0.400000D 01   0.141420D 01    0.200000D 01   0.100000D 01   0.200000D 01      35.4
       2         2   0.140000D 02   0.700000D 01   0.141420D 01    0.200000D 01   0.100000D 01   0.200000D 01      20.2
       3         2   0.800000D 01   0.400000D 01   0.141420D 01    0.200000D 01   0.100000D 01   0.200000D 01      35.4
       4         2   0.110000D 02   0.550000D 01   0.212130D 01    0.450000D 01   0.150000D 01   0.450000D 01      38.6



                                                     TEST ONE

                                                 ADDITIONAL MOMENTS
VARIABLE          MOMENTS ABOUT ORIGIN           MOMENTS ABOUT MEAN       ALPHA 3         ALPHA 4       SKEWNESS        KURTOSIS
                   THIRD          FOURTH        THIRD          FOURTH
       1       0.760000D 02   0.353000D 03   0.0          0.100000D 01   0.0          0.100000D 01   0.0          -0.100000D 01
       2       0.364000D 03   0.269600D 04   0.0          0.100000D 01   0.0          0.100000D 01   0.0          -0.100000D 01
       3       0.760000D 02   0.353000D 03   0.0          0.100000D 01   0.0          0.100000D 01   0.0          -0.100000D 01
       4       0.203500D 03   0.132850D 04   0.0          0.506250D 01   0.0          0.100000D 01   0.0          -0.100000D 01
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) STSUM

pages
1– 2 Identification & Abstract
3– 4 User Instructions
5– 6 I/O
7 Cost—Contents

DESCRIPTIVE TITLE	T Test on Difference between Means
CALLING NAME	(STPAC) TTEST
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

TTEST variables (samples) may be correlated or noncorrelated. Missing observations are permitted. Data may be submitted in the form of a single deck or multiple decks. Output can consist of values of t, frequencies, probabilities, and levels of significance. TTEST will perform a t test on each of all possible pairs of variables, each t being computed independently of all others.

For a t test on the difference between means of correlated samples, the formula used is

$$t = \frac{\Sigma D/N}{\left\{[\Sigma D^2 - \frac{(\Sigma D)^2}{N}]/[N(N - 1)]\right\}^{1/2}},$$

where N is the number of pairs and D is the difference between the scores for each pair. The number of degrees of freedom is expressed by N - 1.

For a t test on the difference between means of noncorrelated samples, with separate variances, the formula used is

$$t = (\overline{X}_1 - \overline{X}_2)/(S_{\overline{X}_1}{}^2 + S_{\overline{X}_2}{}^2)^{1/2},$$

continued

where the X's are the means and the $S_{\overline{X}_1}{}^2$ and $S_{\overline{X}_2}{}^2$ are the squares of the standard errors of the means.

If the frequencies of the two samples are equal, the number of degrees of freedom are equal to the frequency minus 1. If the frequencies are unequal, the program uses the lowest frequency minus 1 as the number of degrees of freedom.

A t test on the difference between means of noncorrelated samples, with pooled variances, involves the formula

$$t = (\overline{X}_1 - \overline{X}_2)/[S^2(\frac{1}{K_1} + \frac{1}{K_2})]^{1/2},$$

where S^2 is the variance—i.e., the sum of squares has been calcalculated by summing the squared deviations for each individual case from the mean of the group in which that case is found—and the K's are the individual group frequencies. The number of degrees of freedom is equal to $K_1 + K_2 - 2$.

REFERENCES

Wert, J.E., Neidt, C.O., and Ahmann, J.S., *Statistical Methods in Educational and Psychological Research* (Appleton–Century–Crofts, New York, 1954), pp.128–142.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential to TTEST. Begin- and End-Data Cards also are necessary if the data have already been read for a previous STPAC program. All other STPAC Control Cards are optional. There are in addition some required and optional Control Cards specific to TTEST; these are described below.

Correlated or Noncorrelated Card (one required)

Columns	*Contents*	
1–10	CORRELATED or	one of these
1–13	NONCORRELATED or	
1–20	NONCORRELATED POOLED	

This card informs the program as to whether the variables are correlated or uncorrelated (noncorrelated), and, if the latter, whether pooled variance or unpooled variance is to be used.

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card should be included only if the values of t are to be punched on cards in addition to being printed.

Decks Card (optional)

Columns	*Contents*	
1– 7	DECKS M or	one of these
1– 5	DECKS	

continued

This card should be included if the data should be submitted in multiple decks rather than in a single deck. DECKS M is used if all decks have equal numbers of variables punched on them and M represents that number of variables. DECKS is used when decks have different numbers of variables punched on them; in this case, the number of variables per card is included in the data. See Description of Input Data, below.

Probabilities Card (optional)

Columns	*Contents*
1–13	PROBABILITIES

This card causes a table of probabilities for each t to be printed.

Significance-Levels Card (optional)

Columns	*Contents*
1–19	SIGNIFICANCE LEVELS

Include this card to print out a table containing entries that describe significance at the 0.05, 0.025, 0.01, and 0.001 levels.

Missing-Data Card (optional)

Columns	*Contents*
1–12	MISSING DATA
14–21	Any number up to eight digits, including a decimal point, that will be read as a code for missing data.

Include this card only if some data are missing. If this card is omitted, it will be assumed that all fields contain data (blanks will be read as zero). If this card is included and Cols. 14–21 are blank, zeros and blanks in the Data Cards will be treated as missing data. If Cols. 14–21 are punched, the number x in a data field will be treated as a case of missing data.

Example (starting in Col. 1)

```
MISSING DATA 99.9
```

This would provide for a case where Data Cards can contain instances of zero observations and instances of missing data. All data fields previously blank would then be coded 99.9 on the Data Cards.

Description of Input Data

Data must be punched according to the format given on the Format

continued

Card. In standard form, each card should contain case and sequence numbers in I format (but only if required for multiple cards per case, as described in the STPAC writeup), followed by data points for each variable in F or E format.

If the DECKS M option (1st form) is used, each card should contain a deck-identification number in I format and a data point in F, E, or D format for each variable in that deck.

If the DECKS option (2nd form) is used, each card should contain not only a deck-identification number but also the number of variables on that card (i.e., that deck), both in I format, in addition to a data point in F, E, or D format for each variable in that deck. All cards for one deck—containing the same deck-identification number and the same number of variables on every card—must be together. Correlated variables (samples) will not be handled with the DECKS option.

The number of a variable is determined by the order in which it is read, which is determined by the format supplied (and the order of the decks if multiple decks are used). In the case of decks, where the first deck contains three variables and the second contains four, the variables will be renumbered from 1 through 7.

Description of Output

Values of t will be printed and also punched at the user's option. In the case of correlated samples, a table of frequencies also will be printed. Tables of probabilities of significance levels will be printed at the user's option.

All tables are printed as triangular matrices. Every variable is numbered and named according to the Variable-Name(s) Card if it is included.

REFERENCES

Wert, J.E., Neidt, C.O., and Ahmann, J.S., *Statistical Methods in Education and Psychological Research* (Appleton—Century—Crofts, New York, 1954), pp.128—142.

SAMPLE INPUT

```
EXECUTE TTEST
VARIABLES 2 SEQUENCED 1 PERCENT
FORMAT(2I4,F4.0)
TITLE TEST OF SECOND [illegible]
FORMAT(2I4,F4.0)
NONCORRELATED
BEGIN DATA
   1   1 175
   1   2 142
   2   1 132
   2   2 411
   3   1 218
   3   2 447
   4   1 151
   4   2 262
   5   1 200
   5   2 402
   6   1 219
   6   2 145
   7   1 234
   7   2 253
   8   1 149
   8   2 199
   9   1 187
   9   2 236
  10   1 123
  10   2 216
  11   1 248
  11   2 211
  12   1 206
  12   2 176
  13   1 179
  13   2 249
  14   1 206
  14   2 214
END DATA
STOP
```

FILMED FROM BEST AVAILABLE COPY

SAMPLE OUTPUT

```
                                          TEST OF SECOND TTNSE DECK

  VARIABLE       NO.OBS.          MEAN   STANDARD ERROR SQUARED
        1          14      0.18764D 03      0.10363D 03
        2          14      0.23593D 03      0.21050D 03




                                          TEST OF SECOND TTNSE DECK

                                                T TEST

                                           UNCORRELATED SAMPLES


              1


    NO.OBS.  14
  2    14   2.724
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) TTEST

pages	
1– 2	Identification & Abstract
3– 5	User Instructions
7– 8	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	Pearson Product-Moment Correlation Coefficient
CALLING NAME	(STPAC) PPMCR
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	William H. Verity Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

PPMCR computes not only a Pearson product-moment correlation coefficient for each possible pair of input variables, but also the mean and standard deviation of each input variable.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

PPMCR can have an unlimited number of observations per variable. The output can be punched and then submitted at a later date as input to another program, such as regression analysis, or it can be passed directly to another program in STPAC. All computations use double-precision arithmetic.

Control Cards and Their Functions

PPMCR has one optional Control Card in addition to those applicable to all STPAC programs. There is *no standard format* for this program, so a Format Card must be supplied. Also, a *Variables Card is required.*

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card will cause PPMCR to punch a deck that can be used later as input to another STPAC program [for example, DNREG, EIN No. 000 0051(h)]. Since intermediate results can be passed directly to another STPAC program, the use of punched-card output is recommended only where it is desirable to break up the run.

Description of Input Data

Data Cards must be supplied to PPMCR, along with the proper format. If, as discussed in the STPAC writeup, case and sequence numbers are used (only when all the variables do not fit on one card), the format must allow for them. Data are read by using an F- or E-format code (both for single-precision floating-point numbers). Case and sequence numbers, if used, are read with an I-format code.

Description of Output

The first page of PPMCR output is a table of means and standard deviations for each of the input variables. Then, starting on a new page, the correlation coefficients are printed in the form of a triangular matrix, with the rows and columns identified by variable

continued

number and name if a Variable-Name(s) Card was used.

If punched output was requested, the punched deck contains

(1) Title Card

(2) A set of Variable-Name(s) Cards, if the variables were named

(3) A Begin-Data Card

(4) Cards containing the means, standard deviations, and correlation coefficients and a card with the number of variables and number of observations

(5) An End-Data Card

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193                                  0051(d)
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168
2995199
END DATA
EXECUTE SIGPP
INPUT FROM PPMCR                         0051(e)
LEVEL  .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR
DEPENDENT Y
PARSIMONY                                0051(h)
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y
EXECUTE CANON
INPUT FROM PPMCR                         0051(g)
LEFT Y
RIGHT X1,X2
EXECUTE FANAL                            0051(i)
INPUT FROM PPMCR
FACTORS 3
EXECUTE VARMX
INPUT FROM FANAL                         0051(j)
STOP
```

SAMPLE OUTPUT

```
                                          SNEDECOR EXAMPLE ON PAGE 414
                                        MEANS AND STANDARD DEVIATIONS

                    MEANS      STD. DEV

 1        X1        11.94        10.15
 2        X2        42.11        13.62
 3         Y        81.28        27.00


                                           CORRELATION COEFFICIENTS


                      1          2
                      X1         X2
 2        X2      0.4616
 3         Y      0.6934     0.3545

TOTAL NUMBER OF DATA CARDS PROCESSED:        18               NUMBER OF OBSERVATIONS:          18

NUMBER OF VARIABLES EXAMINED:   3
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.66 + network overhead

CONTENTS—(STPAC) PPMCR

pages	
1	Identification & Abstract
3– 4	User Instructions
5– 6	I/O
7	Cost—Contents

DESCRIPTIVE TITLE	Significance of Pearson Product-Moment Correlation Coefficient
CALLING NAME	(STPAC) SIGPP
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	W.M. Stiteler, III Mathematics Department, The Pennsylvania State University F. Yates Borden Forestry Department, The Pennsylvania State University H.D. Knoble, The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

SIGPP is to be used optionally in conjunction with the correlation programs to determine the significance of correlation coefficients at any desired probability levels. That is, the null hypothesis r = 0 (alternate: r ≠ 0) is tested for each correlation coefficient.

REFERENCES

U.S. Department of Commerce, *Handbook of Mathematical Functions* (U.S. Government Printing Office, Washington, D.C., 1964), p.949, formula 26.7.8.

Snedecor, G.W., *Statistical Methods* (Iowa State University Press, Ames, Ia., 1961), p. 190.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

In addition to Control Cards specified for STPAC, the following Control Card may be used for this program.

Level Card(s) (optional)

Columns	*Contents*
1–80	LEVEL RRRRRRRR CCCC

There may be from zero to five of these cards for a problem. LEVEL is a key word. RRRRRRRR is a two- to eight-character floating-point field, including a decimal point (e.g., .001 or .05), indicating the probability of making a type I error (rejecting the null hypothesis when it is true). 0 < RRRRRRRR < 1. CCCC is a one- to four-character symbol to be associated with RRRRRRRR (e.g., * or .05).

Description of Input Data

Punched-card output from a correlation program may be used as data. That is,

```
BEGIN DATA
  :
Output from a correlation program
  :
END DATA
```

By using the Input-From Card as described in the STPAC writeup, SIGPP may be executed during the same run as a correlation program.

To repeat a problem using the same data but different Level-Card options, the Same-Data Card may be used according to the STPAC rules.

NOTES:

1. Of the above three options (Punch Card, Input-From Card, Same-Data Card), one must be chosen for a problem.

continued

2. Although means and standard-deviations cards are not necessary for SIGPP, these may be left intact on punched-card input as they will automatically be ignored by this program.

3. Level Card(s), if used, may be placed anywhere after the Execute Card and before the Begin-Data Card if the latter card is present.

The approximations in the program begin to lose accuracy as the number of observations becomes less than 10. According to Ref. 1, the approximations in SIGPP are valid also for Spearman rank coefficients for cases where there are more than eight ranks.

Description of Output

A probability of making a type I error is computed for each correlation coefficient, using a two-tailed test based on a transformation and approximation of the Z (standard normal) probability density function.

If Level Card(s) have been used, the null hypothesis $H_0:r = 0$ is tested versus the alternate $H_0:r \neq 0$ at the specified level(s) of significance and a triangular matrix of results is printed, using corresponding symbols. The sign of the correlation coefficient also is printed with each respective symbol. Thus, each position of the triangular matrix of correlations will have a sign and a symbol that designate the sign and significance of the correlation coefficient of that corresponding position. All probabilities greater than the greatest probability on all of the Level Card(s) appear in the output as #NS#, meaning that the null hypothesis is to be accepted at the level of significance greater than or equal to the largest probability specified.

If no Level Card(s) have been used for a problem, each position of the triangular matrix will have a sign and the actual probability (of making a type I error or rejecting the null hypothesis when it is true) associated with the correlation coefficient of that corresponding position.

REFERENCES

1. Snedecor, G.W., *Statistical Methods* (Iowa State University Press, Ames, Ia., 1961), p. 190.

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193                                   0051(d)
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168.
2995199
END DATA
EXECUTE SIGPP                             0051(e)
INPUT FROM PPMCR
LEVEL  .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR                          0051(h)
DEPENDENT Y
PARSIMONY
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y                               0051(g)
EXECUTE CANON
INPUT FROM PPMCR
LEFT Y
RIGHT X1,X2                               0051(i)
EXECUTE FANAL
INPUT FROM PPMCR
FACTORS 3
EXECUTE VARMX                             0051(j)
INPUT FROM FANAL
STOP
```

SAMPLE OUTPUT

```
SIGNIFICANCE OF CORRELATION COEFFICIENTS                 OS/360 VERSION, OCTOBER, 1967.


                              SNEDECOR EXAMPLE ON PAGE 414

THE NULL HYPOTHESIS: R .EQ. 0 .VS. THE ALTERNATE: R .NE. 0 IS TESTED FOR EACH CORRELATION COEFFICIENT, R.
   PROBABILITIES LESS THAN OR EQUAL TO 0.10000 ARE DESIGNATED BY  TEN INDICATES REJECTION OF THE NULL HYPOTHESIS AT THIS LEVEL.
   PROBABILITIES GREATER THAN 0.10000 ARE DESIGNATED BY     #NS# INDICATES ACCEPTANCE OF THE NULL HYPOTHESIS AT THIS LEVEL.
NOTE: THE SIGN ACCOMPANYING A PROBABILITY OR SYMBOL IS THE SIGN OF THE RESPECTIVE CORRELATION COEFFICIENT.

   NUMBER OF OBSERVATIONS =   18 DEGREES OF FREEDOM =   16


                    1         2
                      X1        X2
   X2     2       + TEN
    Y     3       + TEN     #NS#
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.66 + network overhead

CONTENTS—(STPAC) SIGPP

pages	
1	Identification & Abstract
3– 4	User Instructions
5– 6	I/O
7	Cost—Contents

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

PARCOR can have an unlimited number of observations per variable. Up to 27 variables may be specified on the Include and Delete Cards for the calculation of partial correlations and multiple correlations. All computations are done in double-precision arithmetic.

Control Cards and Their Functions

There is *no standard format* for this program, so a Format Card must be supplied. A Variable Card is also required, and a Variable-Names Card may be included.

PARCOR has four types of optional Control Cards in addition to those applicable to all programs in STPAC.

Print Card (optional)

Columns	*Contents*
1– 5	PRINT

This card instructs the program to print the variance—covariance matrix for the total number of variables.

All Card (optional)

Columns	*Contents*
1– 3	ALL

This card specifies that partial correlations and multiple correlations of all the variables are to be computed, using the entire variance—covariance matrix.

Include Cards (optional)

Columns	*Contents*
1– 7	INCLUDE
9–80	i j k ···

This card instructs the program to compute partial correlation and multiple correlations for variables i, j, k, ···, only. The variable list specified for a single calculation must fit on

continued

one card. The variables may be specified by their number or, if a Variable-Names Card was supplied, by their names. They must be separated by at least one blank, and hence at most 27 variables could be specified for a single calculation.

Delete Card(s) (optional)

Columns	*Contents*
1– 6	DELETE
8–80	i j k ···

This card instructs the program to delete variables i, j, k, ···, from the calculation of partial correlations and multiple correlations. All variables specified for a single calculation must fit on one card. They may be specified by number or by name (if a Variable-Names Card was supplied), and they must be separated by one or more blanks. Up to 27 variables may be specified on each Delete Card.

Example: DELETE AGE SEX WEIGHT HEIGHT

This card specifies that all the variables *except* age, sex, weight, and height should be included in the calculation of partial and multiple correlations. In this example, the variables are specified by name, and it is assumed that a Variable-Names Card was supplied.

NOTE: PARCOR Control Cards may be inserted anywhere among the STPAC Control Cards and they may appear in any order. A given set of Control Cards may consist of any number of Include or Delete Cards and, at most, one Print Card and one All Card for any given set of data.

Description of Input Data

Data Cards must be supplied to PARCOR along with the appropriate Format Card. If, as described in the STPAC writeup, the case and sequence numbers are used (i.e., if all the variables do *not* fit on one card), the format must allow for them. Data are read using F- or E-format code, and the case and sequence numbers, if used, with an I-format code.

Description of Output

If a Print Control Card was supplied, the variance–covariance ance matrix for all the variables is printed in the form of a triangular matrix, together with the means of all the variables. The rows and columns are identified by variable numbers and, if a Variable-Names Card was supplied, by their names.

continued

For each Include, Delete, and All Card, a triangular matrix of partial correlations of the specified variables is printed. The rows and columns are identified by the variable numbers and, if a Variable-Names Card was included, by their names. The heading gives the total number of variables as well as the number of variables included in the calculation of partial correlations and multliple correlations. The multiple correlations of these variables are printed below the matrix.

SAMPLE INPUT

```
EXECUTE PARCOR
VARIABLES 30
VARIABLE NAMES 1=VAR1,2=VAR2,3=VAR3,4=VAR4,5=VAR5,6=VAR6,7=VAR7,8=VAR8
VARIABLE NAMES 9=VAR9,10=VAR10,11=VAR11,12=VAR12,13=VAR13,14=VAR14,15=VAR15,
VARIABLE NAMES 16=VAR16,17=VAR17,18=VAR18,19=VAR19,20=VAR20
TITLE TEST OF PARTIAL CORRELATIONS PROGRAM
FORMAT  (2X,78-1.0)
PRINT
INCLUDE  19 20 21 22 23 24 25 26 27 28 29 30
DELETE VAR1 VAR2 VAR3 VAR4 VAR5 VAR6 VAR7 VAR8 VAR9 VAR10 VAR11 VAR12 VAR13
INCLUDE VAR5 VAR10 VAR15
ALL
DELETE VAR1 VAR3 VAR5 VAR7 VAR29 VAR30
DELET 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
BEGIN DATA
01676566465666677767547656576757655566666644776656565656566666657704657655566660366073
0244507157757705704447605556405655266657502232266546655507503254407755675655540640453
030540530570660430440740750740550635570640440750750650550750640450760560750460053
040475340560635545355640554630345644765546744726755560440733646540670560445450053
050456665765755766456565766744770765775545574770760760670645774650650775750555575
0655544464214344744757544444554554365364326664357444245524455646675454543355755425
070250450550450340550470340450460346534045045036034044044053055045024033555053044
080340340440240334344340440320430230240250320240530240430630320437230450340240453
095350234430440330374330345420225140460220320230434244620410320420640330250330042
100360330430450330340330540530245244440330530230540240645650230440644430240340032
110465455655456444644557045454034055755457764436544535546555475446740656660550745043
125540346545455455525054054543047553754651545424666054726544055445065763053054455
                  .          .           .
                  .          .           .
                  .          .           .
46544044444044534755744034446434460337335350340346546645664355566640145557550505544
473456530450540560430545450430537540454535540637460455450470445567356455534030046
480635230340530554433442534405443304444304304505304503555440345254340440340340240054
490644330330450220334240425540320440430310140546644033444434044034034443304303203034
500445330447540440230540405435446530230530420440336440360430330457550330457033
515445346440540540335440434526344344430430440620455460345555445535340550330330533
520655644440545554435450340540445344544445505455554504403545454056055453030044
530554536465540650445465230550340544440440340430550464340340440530554544444535434
540330434440330140434450340340430335440340330430554530350440530550430420250440035
550560640460540240434545520540340540540460640560550470350550440670560550566350065
567636445755733754444541642711642623553734572464446325626335245444633032645663365642
575635464435445430545440220314416535730354440455434625230345445230420444335545533
5804303405606604305445244303404404254205443404304446420520340540112044444420620024
590410210210410120315110410210220420333235230110210130120115437510410320210130131
604550355555440545540430430450434434534535454444555335620557450256356555230534644
615655340476646554664565550534430346535660440545544545566704473553505404444455555
620570455460755654555645675740540655565670760470670570775671564565555777475044576
635530334560620534445065465463044073442554046056554544665665405661545544304451452
640555334446440540340230540434435545530444650530450450355554350340550540340440063
650120230230120440374220225255220330240140434130225230210220436335420224235440024
660120535330120120210150230250220420110230230520220130210236130220420220330220022
670210426335324220120260215240335414414430145540320315335235244210330315416425022
680225125210127216327236125137124116214220117110110215110127310120415114125110012
690514230140220210120320130220110220320130325510120220330330120220420120120230011
700120120230220210110110110120110110120130210240110110130120110210410110117240011
710440240650570670550630570550640740530450560545640570470240360570370745640440066
END DATA
```

FILMED FROM BEST AVAILABLE COPY

000 0051(f)

SAMPLE OUTPUT

```
                                 TEST OF PARTIAL CORRELATIONS PROGRAM


                                 *** VARIANCE-COVARIANCE MATRIX ***


                  1        2        3        4        5        6        7        8        9        10       11       12
                  VAR1     VAR2     VAR3     VAR4     VAR5     VAR6     VAR7     VAR8     VAR9     VAR10    VAR11    VAR12
  1      VAR1   7.21D 00
  2      VAR2   5.86D-01 2.00D 00
  3      VAR3   2.00D-01 6.43D-01 2.00D 00
  4      VAR4   2.32D 00 1.17D 00 8.29D-01 7.07D 00
  5      VAR5   1.11D 00 6.00D-01 7.71D-01 7.43D-01 2.00D 00
  6      VAR6   1.06D 00 3.43D-01 4.71D-01 1.19D 00 5.29D-01 2.00D 00
  7      VAR7   3.82D 00 4.00D-01 1.43D-01 2.05D 00 1.17D 00 8.00D-01 6.77D 00
  8      VAR8   7.29D-01 3.86D-01 8.86D-01 3.86D-01 8.14D-01 5.86D-01 8.57D-01 2.00D 00
  9      VAR9  -3.71D-01 7.14D-01 8.43D-01-7.14D-02 5.14D-01-1.71D-01 4.29D-02 1.20D 00 2.00D 00
 10      VAR10  2.20D 00 5.43D-01 5.00D-01 2.58D 00 7.57D-01 1.16D 00 2.71D 00 1.33D 00 7.00D-01 7.18D 00
 11      VAR11  2.00D-01 8.00D-01 1.00D 00 7.00D-01 4.43D-01 2.00D-01-1.43D-01 1.11D 00 1.27D 00 8.71D-01 2.00D 00
 12      VAR12  0.0      5.57D-01 5.86D-01-3.14D-01 4.86D-01 1.43D-01-2.29D-01 1.13D 00 1.03D 00 5.29D-01 1.04D 00 2.00D 00
 13      VAR13  2.55D 00 9.43D-01 5.57D-01 3.39D 00 9.00D-01 1.01D 00 2.47D 00 3.00D-01 1.43D-02 2.43D 00 2.43D-01 2.57D-01
 14      VAR14  7.43D-01 7.43D-01 8.86D-01 1.41D 00 4.86D-01 9.86D-01 4.29D-01 8.86D-01 5.86D-01 1.11D 00 1.01D 00 5.57D-01
 15      VAR15  8.71D-01 8.00D-01 8.86D-01 8.29D-01 9.43D-01 7.43D-01 5.00D-01 1.17D 00 8.57D-01 1.29D 00 9.43D-01 8.57D-01
 16      VAR16  2.51D 00 2.71D-01 6.86D-01 2.69D 00 5.71D-01 1.47D 00 2.27D 00 4.43D-01-1.14D-01 1.83D 00 4.57D-01 2.86D-02
 17      VAR17  1.06D 00 6.29D-01 6.57D-01 7.71D-01 6.43D-01 1.03D 00 4.00D-01 7.43D-01 3.14D-01 8.29D-01 4.71D-01 6.29D-01
 18      VAR18  1.06D 00-1.57D-01 4.57D-01 8.57D-02 8.57D-02 2.14D-01 2.00D-01 3.57D-01 2.71D-01 2.57D-01 3.29D-01 3.71D-01
 19      VAR19  1.50D 00 3.86D-01 3.71D-01 2.77D 00 2.29D-01 5.86D-01 1.69D 00-3.57D-01-2.86D-01 8.67D-01 2.86D-01-6.71D-01
 20      VAR20  6.57D-01 9.57D-01 5.86D-01 8.29D-01 7.29D-01 6.29D-01 2.29D-01 1.30D 00 1.31D 00 8.71D-01 1.40D 00 1.03D 00
 21             5.86D-01 0.0      6.86D-01 8.00D-01 8.86D-01 6.86D-01 1.03D 00 6.71D-01 3.43D-01 1.03D 00 5.14D-01 3.86D-01
 22             2.93D 00 2.14D-01 2.71D-01 2.80D 00 5.14D-01 1.13D 00 2.02D 00-1.57D-01-1.00D-01 1.88D 00 2.00D-01-5.29D-01
 23             1.29D-01 7.00D-01 9.43D-01 2.00D-01 6.29D-01 3.86D-01-2.57D-01 1.10D 00 9.43D-01 9.71D-01 1.13D 00 9.14D-01
 24             1.27D-01 6.86D-01 1.04D 00 4.23D-01 6.86D-01 4.29D-01 2.76D-01 1.01D 00 9.71D-01 2.74D-01 7.86D-01 1.13D 00
 25             3.22D 00-2.57D-01 1.71D-01 2.12D 00 7.14D-02 1.03D 00 3.35D 00 6.57D-01-1.57D-01 2.97D 00 4.71D-01 4.00D-01
 26             2.14D-01 8.57D-01 1.20D 00 7.14D-01 7.57D-01 5.29D-01 2.86D-02 9.86D-01 9.86D-01 5.57D-01 1.53D 00 9.43D-01
 27             6.43D-01-1.14D-01 2.57D-01 8.57D-02 4.86D-01 4.71D-01-2.71D-01 1.57D-01 1.43D-02 2.86D-01 1.71D-01 3.57D-01
 28             3.09D 00 4.57D-01-5.71D-02 3.55D 00 1.71D-01 1.01D 00 2.69D 00-1.86D-01-7.43D-01 2.31D 00-4.00D-01-6.00D-01
 29             1.11D 00 5.71D-01 8.00D-01 1.06D 00 1.17D 00 1.09D 00 7.71D-01 1.10D 00 5.71D-01 1.14D 00 9.00D-01 7.43D-01
 30             7.57D-01 2.86D-01 7.57D-01 2.57D-01 1.04D 00 6.86D-01 6.00D-01 1.03D 00 5.43D-01 8.14D-01 6.57D-01 8.71D-01


MEANS           2.23D 00 4.00D 00 4.00D 00 2.39D 00 4.00D 00 4.00D 00 2.83D 00 4.00D 00 4.00D 00 2.04D 00 4.00D 00 4.00D 00
                                   .                                .                                .
                                   .                                .                                .
                                   .                                .                                .
```

continued

000 0051(f)

```
                              *** PARTIAL CORRELATION COEFFICIENTS MATRIX ***

                                    TOTAL NUMBER OF VARIABLES= 30
                                  NUMBER OF VARIABLES IN CORRELATION = 12


                  19        20        21        22        23        24        25        26        27        28        29        30
                 VAR19     VAR20
19    VAR19     1.0000
20    VAR20     0.1128    1.0000
21             -0.0903    0.0308    1.0000
22              0.2069   -0.0282   -0.1084    1.0000
23             -0.1580    0.1176   -0.4258   -0.1023    1.0000
24             -0.0654    0.2735   -0.0791   -0.0657    0.0935    1.0000
25              0.2103   -0.0288    0.2759    0.2955    0.2212   -0.0075    1.0000
26              0.1642    0.3904    0.3189    0.0246    0.4237    0.1164   -0.0762    1.0000
27              0.1327   -0.0018    0.1536    0.0121   -0.0945    0.0584   -0.0725   -0.0143    1.0000
28              0.0778    0.0167   -0.0360    0.3163   -0.0438   -0.0963    0.1972   -0.0392    0.0203    1.0000
29             -0.0226    0.2430    0.2244    0.1581    0.1645    0.1695   -0.0849   -0.0813    0.0788    0.0908    1.0000
30             -0.0800   -0.0135    0.2667    0.1517    0.3432    0.1439   -0.0179   -0.0589    0.2429   -0.0053    0.2324    1.0000



                              *** MULTIPLE CORRELATION COEFFICIENTS ***
           19   0.5104     20   0.8037       21   0.6571      22   0.6413      23   0.8048      24   0.7266
           25   0.5583     26   0.8045       27   0.4939      28   0.5492      29   0.7576      30   0.7618
```

continued

000 0051(f)

000 0051(f)

```
                       *** PARTIAL CORRELATION COEFFICIENTS MATRIX ***

                              TOTAL NUMBER OF VARIABLES= 30
                         NUMBER OF VARIABLES IN CORRELATION = 17


                14        15        16        17        18        19        20        21        22        23        24        25
              VAR14     VAR15     VAR16     VAR17     VAR18     VAR19     VAR20
14   VAR14   1.0000
15   VAR15   0.2418    1.0000
16   VAR16   0.0950   -0.1087    1.0000
17   VAR17   0.0126    0.2085    0.0955    1.0000
18   VAR18   0.1739   -0.1374    0.0372    0.1526    1.0000
19   VAR19  -0.0216   -0.0609    0.2316    0.1186    0.0026    1.0000
20   VAR20   0.2717    0.0304   -0.0381    0.0722   -0.0168    0.1070    1.0000
21           0.0316   -0.2348   -0.0291    0.2704   -0.2352   -0.1155    0.0258    1.0000
22           0.0289    0.0267    0.1951    0.1137   -0.1063    0.1335   -0.0443   -0.1341    1.0000
23           0.0200   -0.1912   -0.0772    0.1776   -0.0262   -0.1632    0.1072   -0.4610   -0.0963    1.0000
24          -0.0772    0.2285    0.2123    0.0547    0.1373   -0.1106    0.2312   -0.0169   -0.1083    0.1215    1.0000
25          -0.0296    0.1327    0.3199   -0.1868    0.2315    0.1263   -0.0170    0.3358    0.2303    0.2539   -0.1079    1.0000
26           0.0884    0.1613    0.0404   -0.2664    0.1082    0.1819    0.3297    0.3871    0.0450    0.4555    0.0801   -0.1458
27           0.0687    0.0550    0.1055    0.0251    0.2056    0.0876   -0.0384    0.1828   -0.0042   -0.0788   -0.0156   -0.1473
28           0.0512    0.0466    0.0615    0.0647    0.0405    0.0513   -0.0134   -0.0283    0.2793   -0.0401   -0.1286    0.1496
29           0.2257    0.2183   -0.0681    0.2379   -0.1162   -0.0133    0.0842    0.1596    0.0837    0.1446    0.0824   -0.0176
30          -0.2198    0.2740    0.1013   -0.1589    0.1427   -0.0647    0.0347    0.3305    0.1449    0.3778    0.0522   -0.1012



                         PARTIAL CORRELATION COEFFICIENTS MATRIX (CONT.)


                26        27        28        29        30

26           1.0000
27          -0.0425    1.0000
28          -0.0400   -0.0139    1.0000
29          -0.0633    0.0377    0.0326    1.0000
30          -0.1084    0.1994   -0.0062    0.2194    1.0000


                         *** MULTIPLE CORRELATION COEFFICIENTS ***
             14   0.7439      15   0.7855      16   0.6874      17   0.6828      18   0.5425      19   0.5646
             20   0.8244      21   0.7149      22   0.6710      23   0.8177      24   0.7647      25   0.6883
             26   0.9266      27   0.5503      28   0.5626      29   0.8205      30   0.7945

*** INPUT ERROR ***
THE VARIABLE -    VAR29  SPECIFIED ON THE INCLUDE OR DELETE CARD WAS NOT DEFINED ON THE VARIABLE-NAMES CARD

 PROGRAM PROCEEDED TO THE NEXT PROBLEM


 ****** INCORRECT CONTROL CARDS ******
THE KEYWORD   DELET      CANNOT BE RECOGNIZED
PROGRAM PROCEEDED TO NEXT CARD
```

000 0051(f)

000 0051(f)

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 7 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.77 + network overhead

CONTENTS—(STPAC) PARCOR

pages
1– 2 Identification & Abstract
3– 5 User Instructions
7–10 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Canonical Correlation
CALLING NAME	(STPAC) CANON
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	William H. Verity Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

CANON solves successively for the most significant canonical correlation coefficients. At the same time, two sets of weights associated with each pair of canonical variates are computed.

All matrix computations are done in double-precision floating-point arithmetic. The set of criteria and predictor variables is divided into two groups: 1 (lefthand) and r (righthand) variables, the smaller set automatically becoming the righthand variables. The program then uses a variation of the jacobian method to extract r eigenvalues and r eigenvectors (righthand weights). These are then used through further calculations to compute r canonical correlation coefficients and r sets of lefthand weights. For each canonical correlation coefficient, chi square is computed with the aid of the eigenvalues and the parameters N (the number of observations), r, and 1.

Original variables are brought into the canonical-correlation problem by means of Right- and Left-Variables Cards. Computationally, it is irrelevant whether the variables on the left or on the right are considered as criteria or predictor variables. However, to save computer time, the set with the smallest number of variables (always put into the righthand category by the pro-

continued

gram) should be designated as *right.* Any or all of the variables brought into the problem may be named by using Name Cards according to STPAC rules.

The maximum output records for a problem may be estimated as follows. Let R be the number of righthand variables, L be the number of lefthand variables, and $R < L$. Then, the number of records equals $R(16 + L) + 10$.

Given two sets of variables—not necessarily random variables—[X_i (where $i = 1, 2, \cdots, p$, the number of predictor variables, and Y_i (where $i = 1, 2, \cdots, q$, the number of criterion variables] the problem that CANON solves is essentially that of maximizing the correlation between certain members of the two sets, simultaneously reducing the others to zero. Canonical correlation, then, may be defined in one sense as a process by which the relationship between two sets of variables is reduced to its simplest form. In canonical correlation, *both multiple criteria and multiple predictors* can be involved. (Note that, when $q = 1$ and $p > 1$, the problem is equivalent to that of multiple regression.)

For a hypothetical example, let $p = 3$ and $q = 2$. Here, it is possible to extract two canonical correlation coefficients—say R_1 and R_2. There is a total of $p + q = 5$ variables—say X_1, X_2, X_3 (predictors) and X_4, X_5 (criteria). There will be two pairs of canonical variates—say U_1V_1 and U_2V_2. Suppose for R_1 that the weights for the predictor variables are P_{11}, P_{12}, P_{13} and for the criteria variables that the weights are C_{11}, C_{12}. Then, the pair of new variates, the canonical variates, which are multiples of a linear function of the original variables, are

$$U_1 = K_{11}(P_{11}X_1 + P_{12}X_2 + P_{13}X_3),$$

$$V_1 = K_{12}(C_{11}X_4 + C_{12}X_5),$$

where the K's are arbitrary constants of proportionality and U_1 and V_1 are correlated as described by R_1. A similar situation exists for U_2 and V_2, which are correlated as described by R_2. Furthermore, in general all of the U's and V's have zero mean and unit variance; any U is independent of any other U and likewise for the V's; the correlation between any U and V is zero, except for r correlations $R_1, R_2, \cdots, R_r$, which are the correlations between U_1 and V_1, U_2 and V_2, $\cdots$, U_r and V_r.

As far as interpretation is concerned, the same difficulty arises as in factor analysis: that of knowing whether or not the linear functions correspond to anything real or whether they are merely mathematical figments.

continued

REFERENCES

Cooley, W.W., and Lohnes, P.R., *Multivariate Procedures for the Behavioral Sciences* (John Wiley & Sons, Inc., New York, 1964), pp. 35–59.

Borko, H., *Computer Applications in the Behavioral Sciences* (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962), pp. 267–279.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

In addition to Control Cards specified in the STPAC writeup, the following Control Cards may be used for this program. At least one of each is necessary unless specified as optional. The key word and parameters for each card may be punched anywhere on that card, separated by one or more blanks, with the stipulation that the *order* of each field be adhered to as follows.

Right-Variables Card(s)

Columns	*Contents*
1–80	RIGHT III III III ··· III

RIGHT is a key word. III is either (1) a field of one- to three-digit variable numbers of those variables to be used as right variables or (2) a field of one- to eight-character variable names of those variables to be used as right variables. In the first case, III must be a number between one and the number of variables that are brought into a problem. In the second case, III must be a variable name as defined on the STPAC Name Card(s). In both cases, there may be as many of these fields per Right-Variables Card(s) as will fit on the card or as are necessary to define the problem. However, a maximum of 50 right variables is allowed. Numbered and named fields may be mixed on one card and between cards.

Examples:

```
RIGHT     7     11      2     1                   4
RIGHT     OAT    WHEAT      CORN
RIGHT     DBH        VOLUME       PINEHT      HRDWDHT      14     17
```

Left-Variables Card(s)

Columns	*Contents*
1–80	LEFT III III III ··· III

LEFT is a key word. III is defined the same as above, except that its application is to left variables.

continued

142

Examples:

```
LEFT   6   3   5   8   9   10   12   21
LEFT   RELATIVE   ABSOLUTE   'NO. SPOR'
LEFT   13    PROBEL    9    STRESS    STRAIN    10
```

Tolerance Card (optional)

Columns	*Contents*
1–80	TOL RRRRRRRR

This card reassigns or overrides the default tolerance to be used when pivoting through the correlation matrix during inversions. The user is cautioned when changing the tolerance, which is set to 0.01 as a default. The tolerance may not be changed to any value outside the range 10^{-1}–10^{-7} and will be reset to 10^{-2} if an attempt is made to do so.

TOL is a key word. RRRRRRRR is a two- to eight-character floating-point field, including a decimal point, designating the tolerance to be used for the current problem.

Description of Input Data

Punched-card output from a Pearson product–moment correlation program may be used as data. That is,

```
BEGIN DATA
  :
Output from a Pearson product–moment correlation program
  :
END DATA
```

By use of the Input-From Card as described in the STPAC writeup, CANON may be executed during the same run as a correlation program.

To repeat a problem using the same data but different Right-Variables and/or Left-Variables options, the Same-Data Card may be used according to the STPAC rules.

NOTES:

1. Of the three above options (Punch Card, Input-From Card, Same-Data Card), one must be chosen for a problem.

2. Although means and standard-deviations cards are not necessary

continued

for CANON, these may be left intact on punched-card input, as they will be ignored by this program.

3. The cards described above may be placed in any order after the Execute Card and before the Begin-Data Card, if one is used.

Description of Output

The normalized and unnormalized vectors of weights associated with the set of criteria variables and the corresponding vectors associated with the set of predictor variables are printed out with each canonical correlation coefficient. The value of chi square, Wilk's lambda, and the probability of obtaining a greater value of chi square associated with the hypothesis that the predictor variate is unrelated to the criteria variate also is printed with each coefficient.

Other pertinent information such as title, variable names, number of observations, degrees of freedom for chi square, and eigenvalues used in calculation of the weights also will be printed.

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168.
2995199
END DATA
EXECUTE SIGPP
INPUT FROM PPMCR
LEVEL   .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR
DEPENDENT Y
PARSIMONY
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y
EXECUTE CANON
INPUT FROM PPMCR
LEFT Y
RIGHT X1,X2
EXECUTE FANAL
INPUT FROM PPMCR
FACTORS 3
EXECUTE VARMX
INPUT FROM FANAL
STOP
```

000 0051(d)
000 0051(e)
000 0051(h)
000 0051(g)
000 0051(i)
000 0051(j)

SAMPLE OUTPUT

```
CANONICAL CORRELATION                          PSU OS/360 STATISTICAL PACKAGE VERSION - MARCH, 1968.
                                  :   SNEDECOR EXAMPLE ON PAGE 414

 EIGENVALUE  1      0.48231278
 ------------------------------ -------
 SUM OF EIGENVALUES =       0.48231278



NO. VARIABLES IN CORRELATION ARRAY =   3                         NO. OF OBSERVATIONS =   16
WILKS LAMBDA = 0.518                                             CHI-SQUARE =  10.5      WITH     2  DEGREES OF FREEDOM
SQUARED CANONICAL CORRELATION COEFFICIENT =  0.482               CANONICAL CORRELATION COEFFICIENT = 0.694

                                 NORMALIZED    UNNORMALIZED                                          NORMALIZED    UNNORMALIZED
      VARIABLE     VARIABLE      LEFT-HAND      LEFT-HAND                VARIABLE     VARIABLE      RIGHT-HAND     RIGHT-HAND
        NAME        NUMBER        WEIGHTS        WEIGHTS                   NAME        NUMBER        WEIGHTS        WEIGHTS

        ----       -------       -------        -------                    ----       -------       -------        -------
          X1          1           0.998          0.673                        Y          3           1.000          1.000
          X2          2           0.065          0.044
------------------------------------------------------- ------------------------------------------------------------------------
PROBABILITY OF A GREATER VALUE OF CHI-SQUARE =       0.005
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead

= $0.66 + network overhead

CONTENTS—(STPAC) CANON

pages	
1– 3	Identification & Abstract
5– 7	User Instructions
9–10	I/O
11	Cost—Contents

DESCRIPTIVE TITLE	Step-Up and Step-Down Multiple Linear Regression
CALLING NAME	(STPAC) UPREG/DNREG
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Dennis Deaven (programming) The Pennsylvania State University Computation Center M.A. Efroymson (flow charts)
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

UPREG/DNREG computes stepwise multiple-regression equations, adding or deleting one variable at a time until no variables remain to add significantly to the equations or until no significant variables remain in the equation. At each step, the following are printed for those variables currently in the equation.

(1) Regression coefficients

(2) Standard deviation of the regression coefficients

(3) Standard regression coefficients

(4) Partial correlation coefficients

(5) Fraction of explained variance

(6) Multiple correlation coefficient

(7) Regression-equation intercept

(8) Standard deviation of the dependent variable (UPREG only)

(9) F value for the variable entering or leaving the equation (UPREG only)

continued

UPREG

In the step-up procedure, intermediate results are used to give statistical information at each step in the calculation. These intermediate answers are also used to control the method of calculation. A number of intermediate-regression equations are obtained, as well as the final multiple-regression equation. These equations are obtained by adding or deleting one variable at a time, giving the following equations.

$$Y = b(0) + b(1)x(1),$$

$$Y = b'(0) + b'(1)x(1) + b'(2)x(2),$$

$$Y = b''(0) + b''(1)x(1) + b''(2)x(2) + b''(3)x(3)$$

$$\vdots$$

The variable added is the one that makes the greatest improvement in goodness of fit. The coefficients represent the best values when the equation is fitted by using the specific variables included in the equation. An important property of this procedure is that a variable may be indicated to be significant in any early stage and thus enter the regression equation. In addition, after several other variables are added to the equation, the initial variable may be indicated to be insignificant. The insignificant variable will be removed from the regression equation before adding an additional variable. Therefore, only significant variables are included in the final multiple-regression equation.

DNREG

The step-down procedure of DNREG is simpler than the step-up procedure of UPREG. If the parsimony option has been chosen, DNREG eliminates one variable at a time, printing the intermediate results at each step. The variable eliminated is the one that contributes least to the overall significance. Note that, once a variable has been eliminated with the procedure, it is never brought back into the equation.

REFERENCES

Efroymson, M.A., "Multiple Regression Analysis," in *Mathematical Methods for Digital Computers*, A. Ralston and H.S. Wilf, Eds. (John Wiley & Sons, Inc., New York, 1965).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program

---

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute Card is essential. If PPMCR [EIN No. 000 0051(d)] is executed during the same run and Name(s) Card(s) were used, then the Variable-Name(s)-Transfer Card should be used. Note that a Variables Card is not required since PPMCR either passes the number of variables or punches it onto a card to be read by UPREG later on. Some additional required and optional Control Cards are specific to UPREG and DNREG. These are described below.

Dependent-Variable Card (one required)

Columns	*Contents*
1– 9	DEPENDENT
11–14	xxx

This card designates the variable to be used as the dependent variable. xxx is either the name or the number of one of the variables.

IN Variables Card (optional; one or more)

Columns	*Contents*
1– 2	IN
4– . . .	xxx yyy zzz ··· .

This card selects variables to be forced into the equation no matter what their significance. xxx, yyy, and zzz are either the numbers or names of some of the variables.

OUT Variables Card (optional; one or more)

Columns	*Contents*
1– 3	OUT
5– . . .	xxx yyy zzz ··· .

continued

This card selects variables that are *never* wanted in the equation, no matter how significant any might be. xxx, yyy, and zzz are either the numbers or names of some of the variables.

Tolerance Card (optional)

Columns	*Contents*
1– 9	TOLERANCE
11–13	xxx

This card changes the tolerance used in pivoting through the matrix. The default tolerance is 0.01. The correlation matrix is inverted by using a Gauss–Jordan technique and selecting the largest diagonal element at any given stage on which to pivot. If this pivot should become small, it often indicates a degeneracy in the correlation matrix, which indicates that one of the "independent" variables is, in fact, approximately a linear combination of other independent variables. If the tolerance is set too low, the program may develop erroneous results owing to inverting a singular correlation matrix. In any case, the *tolerance must be greater than zero.*

F-Value Card (optional; UPREG only)

Columns	*Contents*
1– 6	FVALUE
8–10	xxx

This card selects the critical F value that the program uses to determine whether to add or delete a variable from the equation. The default F value is 2.5.

Parsimony Card (optional; DNREG only)

Columns	*Contents*
1– 9	PARSIMONY

This card indicates to DNREG that the step-down procedure should be used, eliminating one variable at a time from the equation. If the parsimony option is not selected, only one set of results will be printed for each problem—that is, one equation with all variables present except those declared OUT.

Description of Input Data

Data for UPREG and DNREG are a set of means, standard deviations, and correlation coefficients, along with the number of variables

continued

and number of observations. Normally, the input comes from PPMCR (from a previous problem during the same run using the Input-From PPMCR Card *or* by using a deck that was punched by PPMCR during some previous job). If the input is supplied by cards, they must be identical in format to those punched by PPMCR.

Description of Output

The output from UPREG and DNREG consists of a series of tables, one table being produced each time a variable is added to or deleted from the equation. Each table contains all the information specified in the Functional Abstract.

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193                                  000 0051(d)
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168.
2995199
END DATA
EXECUTE SIGPP                            000 0051(e)
INPUT FROM PPMCR
LEVEL  .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR
DEPENDENT Y                              000 0051(h)
PARSIMONY
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y
EXECUTE CANON                            000 0051(g)
INPUT FROM PPMCR
LEFT Y
RIGHT X1,X2
EXECUTE FANAL                            000 0051(i)
INPUT FROM PPMCR
FACTORS 3
EXECUTE VARMX                            000 0051(j)
INPUT FROM FANAL
STOP
```

SAMPLE OUTPUT—UPREG

```
MULTIPLE REGRESSION ANALYSIS FOR: SNEDECOR EXAMPLE ON PAGE 414

TOLERANCE=.010

TOTAL NUMBER OF VARIABLES =   3
NUMBER OF OBSERVATIONS =     18

FVALUE =     2.50000

DEPENDENT VARIABLE :   3(        Y)

START REGRESSION, STANDARD DEVIATION OF THE DEPENDENT VARIABLE:     3(        Y) =       26.996
```

continued

```
STEP NUMBER      1
   VARIABLE ENTERING:                             1(      X1)
   STANDARD ERROR OF ESTIMATE:                    0.200510 02
   F VALUE:                                       0.148170 02
   INTERCEPT:                                     0.592590 02
   MULTIPLE CORRELATION COEFFICIENT:                  0.69340
   FRACTION OF EXPLAINED VARIANCE:                    0.48081

               INDEP. VARIABLE       REGRESSION       STAND.DEVIATION   STAND.REGRESSION   PARTIAL CORRELATION
               NUMBER AND NAME       COEFFICIENT      FOR COEFFICIENT   COEFFICIENT        COEFFICIENT
                  1        X1        0.184344E 01       0.478902E 00      0.6934031            0.6934

               ANALYSIS OF VARIANCE

SOURCE OF                SUM OF          MEAN          F
VARIATION       DF       SQUARES         SQUARE        RATIO

DUE TO REG       1    5.957020 03    5.957020 03    1.481710 01

ABOUT REG       16    6.432590 03    4.020370 02

TOTAL           17    1.238960 04



                                              DIAGONAL ELEMENTS

          VARIABLE                                  VARIABLE                                  VARIABLE
          IDENTIFICATION      VALUE                 IDENTIFICATION      VALUE                 IDENTIFICATION      VALUE
           1          X1      1.000 ENTR             2          X2      0.787                  3           Y      0.519 DEP
```

SAMPLE OUTPUT—DNREG

```
MULTIPLE REGRESSION ANALYSIS FOR: SNEDECOR EXAMPLE ON PAGE 414

PARSIMONY OPTION SELECTED

TOLERANCE=.010

TOTAL NUMBER OF VARIABLES =   3
NUMBER OF OBSERVATIONS =     18


NUMBER OF VARIABLES ELEGIBLE FOR ELIMINATION BY PARSIMONY:   2
THE NUMBER ASSOCIATED WITH THESE VARIABLES AND THEIR NAMES, IF SPECIFIED, ARE LISTED BELOW.
  1(       X1),  2(       X2)


DEPENDANT VARIABLE(NAME,IF SPECIFIED, AND ASSOCIATED NUMBER):    3(        Y)
```

continued

```
                    INDEP. VARIABLE        REGRESSION         STAND.DEVIATION    STAND.REGRESSION    PARTIAL CORRELATION
                    NUMBER AND NAME        COEFFICIENT        FOR COEFFICIENT    COEFFICIENT         COEFFICIENT
                       1         X1        0.178977D 01         0.556743D 00       0.6732183            0.6387
                       2         X2        0.866492D-01         0.414943D 00       0.0437310            0.0538

NUMBER OF VARIABLES PRESENTLY ELIMINATED:   0

CANDIDATE FOR THE NEXT ELIMINATION-VARIABLE:     2(       X2)
F-RATIO OF VARIANCE OF AGGREGATE ELIMINATED AND CANDIDATE VARIABLES
                         TO UNEXPLAINED VARIANCE WITH NONE ELIMINATED: 0.4361D-01
FRACTION OF EXPLAINED VARIANCE:   0.482
MULTIPLE CORRELATION COEFFICIENT:  0.6945
INTERCEPT  B(0):  0.562510D 02

                    ANALYSIS OF VARIANCE

SOURCE OF                   SUM OF           MEAN            F
VARIATION        DF         SQUARES          SQUARE          RATIO

DUE TO REG        2    5.975670D 03    2.987830D 03    6.987510D 00

ABOUT REG        15    6.413940D 03    4.275960D 02

TOTAL            17    1.238960D 04
```

continued

```
               INDEP. VARIABLE       REGRESSION      STAND.DEVIATION   STAND.REGRESSION   PARTIAL CORRELATION
               NUMBER AND NAME       COEFFICIENT     FOR COEFFICIENT   COEFFICIENT        COEFFICIENT
                  1       X1         0.184344D 01      0.478902D 00      0.6934031           0.6934

NUMBER OF VARIABLES PRESENTLY ELIMINATED:   1
THE NUMBER ASSOCIATED WITH THE ELIMINATED VARIABLES AND THEIR NAMES, IF SPECIFIED, ARE LISTED BELOW.
  2(       X2)

CANDIDATE FOR THE NEXT ELIMINATION-VARIABLE:    1(      X1)
F-RATIO OF VARIANCE OF AGGREGATE ELIMINATED AND CANDIDATE VARIABLES
                    TO UNEXPLAINED VARIANCE WITH NONE ELIMINATED: 0.6988D 01
FRACTION OF EXPLAINED VARIANCE:   0.481
MULTIPLE CORRELATION COEFFICIENT:  0.6934
INTERCEPT  B(0):  0.592590D 02

               ANALYSIS OF VARIANCE

SOURCE OF               SUM OF           MEAN            F
VARIATION      DF       SQUARES          SQUARE        RATIO

DUE TO REG      1   5.957020 03   5.957020 03   1.481710 01

ABOUT REG      16   6.432590 03   4.020370 02

TOTAL          17   1.238960 04

VALUE OF DETERMINANT =      0.7870D 00
AVERAGE ABSOLUTE DEVIATION FROM ORIGINAL ELEMENTS OF INVERSE-INVERSE MATRIX ELEMENTS =    0.20046D-16
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.66 + network overhead

CONTENTS—(STPAC) UPREG/DNREG

pages

1– 2 Identification & Abstract
3– 5 User Instructions
7–12 I/O
13 Cost—Contents

DESCRIPTIVE TITLE	Factor Analysis (or Principal-Components Analysis)
CALLING NAME	(STPAC) FANAL
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	J. Cooley D. Laird L. Pryor J. McConnochie The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

FANAL solves successively for the most dominant factors represented in a (symmetric) correlation matrix. Factors are ranked according to the variance accounted for. The number of factors to be extracted is specified in advance. The program may also be used to find the dominant eigenvalues and eigenvectors of other (symmetric) positive definite matrices.

All computations are done in double-precision floating-point arithmetic. The method used is an iterative one in which a vector converges to the eigenvector. The largest eigenvalue is found first and then a deflation process is used to find the successively smaller eigenvalues. An extrapolation procedure is used to accelerate convergence of the basic iterative method.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Description of Input

The input to FANAL consists of the Control Cards described below and the lower half of the symmetric matrix plus optional diagonal elements. Normally, the symmetric matrix is the output from PPMCR [EIN No. 000 0051(d)], another STPAC program, which computes the Pearson product-moment correlation coefficients for a set of variables. The output from PPMCR may be in the form of punched cards received from some previous job or it may be passed directly to FANAL by the Input-From PPMCR option.

The symmetric matrix may also come from another program in STPAC or from a user's program. At present, PHICO [EIN No. 000 0051(k)], SPRHO [EIN No. 000 0051(m)], and PPMCR can pass their results to FANAL. In the case of a user's program, the data must be punched in *exactly* the same format that PPMCR punches its output.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute Card is required. If the data were punched by the user, a Variables Card is required also.

Factors Card (one or none)

Columns	*Contents*
1– 7	FACTORS
9–80	n, the number of factors to be extracted from the matrix

If this card is omitted, the program will attempt to extract as many factors as there are variables so long as the variance accounted for is greater than or equal to the minimum variance.

Minimum-Variance Card (one or none)

Columns	*Contents*
1–16	MINIMUM VARIANCE
18–80	x, any floating-point number, with the decimal point punched

continued

x indicates the eigenvalue at which factor extraction will be terminated. If this card is not supplied, the default minimum variance will be zero.

Correlation Card (one or none)

Columns	*Contents*
1–18	CORRELATION MATRIX

This card is a request that the input matrix be printed out.

Residual Card (one or none)

Columns	*Contents*
1–15	RESIDUAL MATRIX

This card is a request that the residual matrix be printed out prior to termination of the program.

Diagonal Card (one or none)

Columns	*Contents*
1– 8	DIAGONAL
10–80	d, a floating-point number, with the decimal point punched, followed by an optional E or D field

d is the value to be placed into each diagonal element of the matrix. An alternative version permits the entry of unique diagonal elements.

Columns	*Contents*
1–14	DIAGONAL CARDS

This card indicates that the diagonal elements will be read from Data Cards following the Begin-Data Card. They must be punched according to a specific format.

Columns	*Contents*
1	Blank
2	D
3	Blank
4– 6	Card sequence number
7–18	1st diagonal element on this card
⋮	⋮
67–78	6th diagonal element on this card

continued

Thus, card 1 would contain the diagonal elements (1,1) through (6,6) and card 5, for example, would contain the diagonal elements (25,25) through (30,30). Note that this format is identical to that used by PPMCR to punch its means and standard deviations, except for the letter D, which designates these cards as diagonal cards.

Description of Output

For each factor, the output includes the variance accounted for (the eigenvalue λ_i), the eigenvector U_i, and the vector of factor loadings f_i, where $f_i = U_i\sqrt{\lambda_i}$.

At the user's option, the input matrix can be printed out and, at the completion of the factor extraction, the residual matrix can be printed out. Also, a punched deck can be requested to be entered at a later time to VARMX [EIN No. 000 0051(j)].

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193                                  0051(d)
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168.
2995199
END DATA
EXECUTE SIGPP
INPUT FROM PPMCR                         0051(e)
LEVEL  .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR
DEPENDENT Y
PARSIMONY                                0051(h)
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y
EXECUTE CANON
INPUT FROM PPMCR                         0051(g)
LEFT Y
RIGHT X1,X2
EXECUTE FANAL                            0051(i)
INPUT FROM PPMCR
FACTORS 3
EXECUTE VARMX                            0051(j)
INPUT FROM FANAL
STOP
```

SAMPLE OUTPUT

```
PRINCIPAL COMPONENTS ANALYSIS - SEPTEMBER 1968 VERSION - THE PENNSYLVANIA STATE UNIVERSITY COMPUTATION CENTER.

PROBLEM NAME:                        SNEDECOR EXAMPLE ON PAGE 414


THIS PROBLEM HAS   3 VARIABLE(S) AND   3 FACTOR(S).



   FACTOR                        TEST      FACTOR       VARIANCE        EIGENVECTOR    ITERATIONS
   NUMBER                       NUMBER     LOADING    (EIGENVALUE)                     REQUIRED
     1           X1               1        0.89601      2.02068          0.63032          10
                 X2               2        0.70114                       0.49324
                  Y               3        0.85220                       0.59951


   FACTOR                        TEST      FACTOR       VARIANCE        EIGENVECTOR    ITERATIONS
   NUMBER                       NUMBER     LOADING    (EIGENVALUE)                     REQUIRED
     2           X1               1        0.18620      0.68498   ---    0.22498          10
                 X2               2       -0.70775                      -0.85515
                  Y               3        0.38653                       0.46702


   FACTOR                        TEST      FACTOR       VARIANCE        EIGENVECTOR    ITERATIONS
   NUMBER                       NUMBER     LOADING    (EIGENVALUE)                     REQUIRED
     3           X1               1        0.40311      0.29433          0.74302          3
                 X2               2       -0.08653                      -0.15950
                  Y               3       -0.35263                      -0.64999
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.66 + network overhead

CONTENTS—(STPAC) FANAL

pages	
1	Identification & Abstract
3– 5	User Instructions
7– 8	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	Varimax Rotation
CALLING NAME	(STPAC) VARMX
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	J. Cooley D. Thompson John McConnochie The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

VARMX can perform a number of orthogonal rotations on an arbitrary matrix of factor loadings, using the normal varimax criterion. The result is a unique (within tolerance limits) matrix of factor loadings.

All computations are done in double-precision floating-point arithmetic. First, the original factor loadings as produced by FANAL [EIN No. 000 0051(i)] are normalized by dividing the loadings in each row by the square root of the communality (sum of squares) for that row. The rotation is performed with a criterion of .005 and the resulting loadings are denormalized to give the rotated matrix.

REFERENCES

Kaiser, H.F., "Computer Program for Varimax Rotation in Factor Analysis," Educ. Psychol. Meas. *19*, 413–420 (1959).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Description of Input

The input to VARMX consists of the Control Cards described below and the matrix of factor loadings to be rotated. The factor matrix is the output from FANAL [EIN No. 000 0051(i)], either passed directly to VARMX by using the Input-From FANAL option or in the form of a card deck punched by FANAL during some previous job.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute Card is required and either an Input-From FANAL Card or a Begin-Data/End-Data sequence with a deck punched by FANAL. There are some additional Control Cards specific to VARMX. They are discussed below.

Rotate Card(s) (none, one, or more)

Columns	*Contents*
1– 6	ROTATE
8–80	$n_1 \cdots n_k$

This general form of the Rotate Card requests that k distinct rotations be performed, first rotating the first n_1 factors, then the first n_2 factors, and finally the first n_k factors. If this card is omitted, the entire matrix of factor loadings will be rotated.

NOTE: When VARMX is completed, the output from FANAL is no longer available. It is destroyed after the last rotation request is completed.

Print Card(s) (one or more)

Columns	*Contents*
1– 5	PRINT

This card requests a printout of the input matrix.

continued

Punch Card(s) (one or more)

Columns	*Contents*
1– 5	PUNCH

This card requests a punched deck of the matrix of rotated factor loadings. For each rotation, a set of cards will be punched for each variable, with each set of cards containing the corresponding rotated factor loadings for that variable. The format of the punched cards is ('VAR',I3,'CD',I2,7F10.5). The first integer field is the variable number, the second is a card sequence number for this variable, and the seven floating-point fields are for the rotated loadings. This deck can then be submitted to PPMCR [EIN No. 000 0051(d)] to correlate the rotated factors. Actually, a Format Card is punched along with some other Control Cards so that about all that the deck needs to submit it to PPMCR is an Execute PPMCR Card.

Description of Output

For each rotation, the output includes the communalities and the rotated matrix of factor loadings. At the user's option, the original input matrix can be printed out and a deck of cards containing the rotated matrix can be obtained for possible submission to PPMCR at a later date.

SAMPLE INPUT

```
EXECUTE PPMCR
TITLE SNEDECOR EXAMPLE ON PAGE 414
VARIABLES 3
VARIABLE NAMES 1=X1,2=X2,3=Y
FORMAT(F3.1,F2.0,F4.2)
BEGIN DATA
0045364
0042360
0311971
0063461
0472454
0176577
0944481
1013193                          ————0051(d)
1162993
1265851
1093776
2314696
2315077
2164493
2315695
0193654
26858168.
2995199
END DATA
EXECUTE SIGPP
INPUT FROM PPMCR                 ————0051(e)
LEVEL  .1 TEN
EXECUTE DNREG
INPUT FROM PPMCR
DEPENDENT Y
PARSIMONY                        ————0051(h)
EXECUTE UPREG
INPUT FROM PPMCR
DEPENDENT Y
EXECUTE CANON
INPUT FROM PPMCR                 ————0051(g)
LEFT Y
RIGHT X1,X2
EXECUTE FANAL
INPUT FROM PPMCR                 ————0051(i)
FACTORS 3
EXECUTE VARMX
INPUT FROM FANAL                 ————0051(j)
STOP
```

SAMPLE OUTPUT

```
                                                  VARIMAX ROTATION

                                              SNEDECOR EXAMPLE ON PAGE 414


THE INPUT MATRIX FROM FACTOR ANALYSIS CONTAINS     3 VARIABLES AND     3 FACTORS.
                                             ROTATION OF FIRST   3 FACTORS
          COMMUNALITIES

        X1      1       1.00000
        X2      2       1.00000
         Y      3       1.00000



                                 THE ROTATED MATRIX OF FACTOR LOADINGS

                     F  1        F  2        F  3
    X1     1        0.37412    -0.23774     0.89639
    X2     2        0.14472    -0.96959     0.19737
     Y     3        0.92506    -0.15712     0.34580
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on central processor unit was 6 seconds, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.66 + network overhead

CONTENTS—(STPAC) VARMX

pages	
1	Identification & Abstract
3– 4	User Instructions
5– 6	I/O
7	Cost—Contents

DESCRIPTIVE TITLE	Phi Coefficient
CALLING NAME	(STPAC) PHICO
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

PHICO processes a maximum of 105 variables and computes phi coefficients for all possible pairs of variables.

Data are dichotomous and are coded 0 and 1 or 1 and 2. The equation used is

$$\text{phi} = (ad-bc)/[(a+b)(c+d)(a+c)(b+d)]^{1/2}.$$

REFERENCES

Wert, J.E., Neidt, C.O., and Ahmann, J.S., *Statistical Methods in Educational and Psychological Research* (Appleton-Century-Crofts, New York, 1954).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program

---

Control Cards and Their Functions

Of the various STPAC Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards *are essential* for PHICO. Begin- and End-Data Cards are also necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional. There are some additional required and optional Control Cards specific to PHICO. These are described below.

Code Card (one required)

Columns	*Contents*
1– 4	CODE
6	A: data punched 0 and 1 B: data punched 1 and 2

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

If this card is included, phi coefficients will be punched on cards in addition to being printed.

Description of Input Data

Data must be punched according to the format given on the Format Card. On each card should be punched a case and sequence number (only if required, as described in the STPAC writeup) and the data points in I format. The "number" of a variable is determined by the order in which it is read, which is determined by the format supplied.

Description of Output

Phi coefficients are printed in a triangular matrix. All variables are numbered. Rows and columns will also be named according to the Variable-Names Card, if included. Card ouput is also produced when requested.

SAMPLE INPUT

```
EXECUTE PHICO
TITLE TEST OF 1S AND 2S
VARIABLES 15
FORMAT(15I1)
CODE B
BEGIN DATA
1211111122221111
212112111122211
121222211121221
121111122211112
121212122211112
12121111122222111
121211112211111222
1212121111112221
END DATA
STOP
```

000 0051(k)

SAMPLE OUTPUT

```
                                        TEST OF 1S AND 2S

                                 PHI COEFFICIENT SYMMETRIC MATRIX

                                        NUMBER OF CASES    8


            1          2          3          4          5          6          7          8          9         10         11         12

-  2  -1.000
-  3   1.000     -1.000
-  4  -0.488      0.488     -0.488
-  5  -0.143      0.143     -0.143      0.293
-  6   0.378     -0.378      0.378      0.258      0.378
-  7  -0.143      0.143     -0.143      0.293      1.000      0.378
-  8  -0.218      0.218     -0.218     -0.149     -0.218      0.0       -0.218
-  9  -0.378      0.378     -0.378     -0.258     -0.378     -0.500     -0.378      0.577
- 10  -0.488      0.488     -0.488     -0.067     -0.488     -0.775     -0.488      0.447      0.775
- 11   0.378     -0.378      0.378     -0.258      0.378      0.0        0.378     -0.577     -0.500     -0.258
- 12   0.488     -0.488      0.488     -0.467     -0.293     -0.258     -0.293     -0.447     -0.258      0.067      0.775
- 13   0.378     -0.378      0.378      0.258      0.378      0.500      0.378     -0.577     -1.000     -0.775      0.500      0.258
- 14  -0.293      0.293     -0.293      0.600      0.488      0.258      0.488     -0.447     -0.775     -0.467      0.258     -0.067
- 15  -0.143      0.143     -0.143      0.293     -0.143      0.378     -0.143     -0.218     -0.378     -0.488     -0.378     -0.293



           13         14

- 14   0.775
- 15   0.378      0.488
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—STPAC

pages
1 Identification & Abstract
3 User Instructions
5– 6 I/O
7 Cost—Contents

DESCRIPTIVE TITLE	Kendall Rand Correlation Coefficients (tau)
CALLING NAME	(STPAC) KETAU
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

KETAU computes Kendall rank-correlation coefficients (tau), the values of z scores corresponding to them, and the probabilities and levels of significance of the z scores, for all possible pairs of variables. A correction is made for ties. The correlation coefficients will also be punched on cards, at the option of the user.

REFERENCES

Siegel, S., *Nonparametric Statistics for the Behavioral Sciences* (McGraw—Hill Book Co., Inc., New York, 1956), pp.213—222.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program

---

Control Cards and Their Functions

Of the various STPAC Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential for KETAU. Begin- and End-Data Cards are also necessary if the data have not been already read for a previous STPAC program. All other STPAC Control Cards are optional. There is one additional optional Control Card specific to KETAU. This is described below.

Punch Card (optional)

Columns	*Contents*
1– 5	Punch

This card should be included if the user wishes to have the values of tau punched, in addition to being printed.

Description of Input Data

Data for all variables should be punched together. The number of a variable is determined by the order in which it is read, which is determined by the format supplied. Thus, the first variable read will be variable 1, the second will be variable 2, and so on. There is no standard format for raw or ranked data, but the data must be in F or E format. Sequence or case numbers, if needed, must be in I format.

Description of Output

The printed output for a KETAU problem consists of four triangular matrices. All are headed by the title that the user supplies. All variables are numbered and both rows and columns will be named according to the Variable-Names Card, if it is included. The first matrix consists of the values of tau for each possible pair of variables. The second is a table of z scores associated with the corresponding coefficients. The third give the probabilities of the z scores and the fourth gives their level of significance. If the total number of observations is less than 10, z scores should be interpreted with caution. However, special tables of the critical values of tau in this range are available.

SAMPLE INPUT

```
EXECUTE KETAU
FORMAT(2F2.0,F4.1)
TITLE KENDALLS RANK CORRELATION SIEGEL  PAGE 227
VARIABLE NAMES 1 = STRIVING, 2 = AUTHORIT 3 = CONFORMITY
VARIABLES 3
BEGIN DATA
 3 2 15
 4 6 15
 2 5 35
 1 1 35
 810 50
11 9 60
10 8 70
 6 3 80
 7 4 90
1212105
 5 7105
 911120
END DATA
STOP
```

SAMPLE OUTPUT

```
                                        KENDALLS RANK CORRELATION SIEGEL  PAGE 227

                                          KENDALL RANK CORRELATION COEFFICIENT

                                              NUMBER OF OBSERVATIONS IS       12

              1         2
           STRIVING  AUTHORIT
AUTHORIT-  2  0.6667
CONFORMI-  3  0.3722    0.3412

                                                      CORRESPONDING Z SCORES

              1         2
           STRIVING  AUTHORIT
AUTHORIT-  2  3.0172
CONFORMI-  3  1.6845    1.5441

                            CORRESPONDING PROBABILITIES OF Z SCORES(PROB (X GREATER THAN OR EQUAL TO Z) )

              1         2
           STRIVING  AUTHORIT
AUTHORIT-  2  0.0013
CONFORMI-  3  0.0460    0.0613

                                                      LEVELS OF SIGNIFICANCE

                                                  BLANK IF   0.050 < P
                                                      * IF   0.025 < P < 0.050
                                                     ** IF   0.010 < P < 0.025
                                                    *** IF   0.001 < P < 0.010
                                                   **** IF           P < 0.001

              1         2
           STRIVING  AUTHORIT
AUTHORIT-  2     ***
CONFORMI-  3       *
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the central processor unit was 4 seconds at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) KETAU

pages	
1	Identification & Abstract
3	User Instructions
5– 6	I/O
7	Cost—Contents

DESCRIPTIVE TITLE	Spearman Rank Correlation Coefficients (rho)
CALLING NAME	(STPAC) SPRHO
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

SPRHO computes Spearman rank-order correlation coefficients (rho) and the values of t corresponding to them for all possible pairs of variables. A correction is made for ties. The correlation coefficients will also be punched on cards at the option of the user.

REFERENCES

Siegel, S., *Nonparametric Statistics for the Behavioral Sciences* (McGraw-Hill Book Co., Inc., New York, 1956), pp. 202-213.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various STPAC Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential for SPRHO. Begin- and End-Data Cards are also necessary if the data have not been already read for a previous STPAC program. All other STPAC Control Cards are optional. There is one additional optional Control Card specific to SPRHO. It is described below.

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card should be included if the user wishes to have the correlation coefficients punched in addition to being printed.

Description of Input Data

Data for all variables within one observation should be punched together. The number of a variable is determined by the order in which it is read, which is determined by the format supplied. Thus, the first variable read will be variable 1, the second will be variable 2, etc. There is no standard format, but the data must be in F or E format. Sequence and case numbers, if needed, must be in I format.

Description of Output

The printed output for a SPRHO problem consists of two triangular matrices. Both are headed by the title that the user supplies. All variables are numbered and both rows and columns will also be named according to the Variable-Name(s) Card, if it is included. The first matrix consists of the Spearman rank-order correlation coefficients for all possible pairs of variables. The second is a table of the values of t associated with the corresponding values of rho. Degrees of freedom are given as part of the heading. If the number of observations is less than 10, these values should be interpreted with caution. However, special tables of critical values of rho in this range are available.

SAMPLE INPUT

```
EXECUTE SPRHO
VARIABLES 21 SEQUENCED 17 PERCARD
VARIABLE NAMES 1=ONE,2=TWO,3=THREE,4=FOUR
TITLE    DEMONSTRATION OF SPRHO
FORMAT(2I2,19F4.2)
BEGIN DATA
 1 1 1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
 1 230   30  10  18
 2 1 3   5   6   8   9   4   2   1   3  28   9   8   6   4   5   2   1   9  10
 2 2235 29   8   85
 3 1 5  21  22  23  24  25  26  27  28  29  30  21  12  13  14  15  16  17  18
 3 2155 19  10  165
 4 1 20  200023  25  26  21  22  20  26  25  24  12  13  15  14  17  22  30  24
 4 226  26  13  205
 5 1 1   2   3   4   5   6   7   8   9   9   8   7   6   5   4   3   2   1   2
 5 2 1   3  26  25
 6 1 9   8   7   6   5   4   3   2   1   1   2   3   4   5   6   7   8   9   5
 6 2 65  1  175 145
 7 1 9   8   7   6   5   4   3   2   1   1   2   3   4   5   6   7   8   9   5
 7 221  205 13  11
 8 1 2  10  23  24  25  30  26  27  28  29  12  11  10  14  15  16  17  18  19
 8 226  10   2   2
 9 1 2  10  23  24  25  30  26  27  28  29  12  11  10  14  15  16  17  18  19
 9 213   4  21  22
10 111  22  31  22  24  25  26  12  13  14  15  16  17  18  19  29  28  27  26
10 2 35 13   5   5
11 111  22  31  22  24  25  26  12  13  14  15  16  17  18  19  29  28  27  26
11 2 65  7  21  23
12 1 2  19  18  17  16  15  14  13  12  11  12  13  14  15  16  17  18  19  11
12 228  28  175 165
13 1 2  19  18  17  16  15  14  13  12  11  12  13  14  15  16  17  18  19  11
13 2235 24  155 12
14 115  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  11  12  13
14 219  145 10   85
15 115  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  11  12  13
15 229  27   35  6
16 122  33  21  22  24  25  26  27  28  29  30  11  10  12  13  14  15  16  17
16 2 35  6  29  27
17 122  33  21  22  24  25  26  27  28  29  30  11  10  12  13  14  15  16  17
17 210   85 19  145
18 111  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
18 2155 12  235 24
19 111  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
19 2175 165 28  28
20 1 9   8   7   6   5   4   3   2   1   1   2   3  13  12   5   5  25  25  35
20 221  23   65  7
21 1 9   8   7   6   5   4   3   2   1   1   2   3  13  12   5   5  25  25  35
21 2 5   5   35 13
22 130  30  30  30  30  30  30  30  30  30  29  29  29  28  28  28  28  27  27
22 221  22  13   4
23 130  30  30  30  30  30  30  30  30  30  29  29  29  28  28  28  28  27  27
23 2 2   2  26  10
24 1 1   1   1   1   1   2   2   2   2   3   3   3   3   4   5   6   5   4  55
24 213  11  21  205
25 12323232323232322323232323232323245454545454546565656565656565653232323232323232323
25 2175 145  65  1
26 12323232323212121212121212112323565655656565656565698989898965654545454545455445
26 226  25   1   3
27 1999999999666666666666655555555552235332323232312121212221212221212121212121212
27 213  205 26  26

28 1353535353535353535353535355565565656262626262989898989846446466466466446446
28 210  165 155 19
29 1353535353535353535353535355565565656262626262989898989846446466466466446446
29 2 8   85 235 29
30 1 1   1   1   1   1   2   2   2   2   3   3   3   3   4   5   6   5   4  55
30 210  18  30  30
END DATA
STOP
```

SAMPLE OUTPUT

```
                                   DEMONSTRATION OF SPRHO

                               SPEARMAN RANK CORRELATION COEFFICIENT

                                 NUMBER OF OBSERVATIONS IS       30

            1          2          3          4          5          6          7          8          9         10         11         12
              ONE        TWO    THREE       FOUR
  TWO-  2  0.865
THREE-  3  0.752      0.906
 FOUR-  4  0.734      0.878      0.952
     -  5  0.720      0.880      0.950      0.991
     -  6  0.739      0.868      0.943      0.982      0.971
     -  7  0.782      0.906      0.950      0.967      0.958      0.989
     -  8  0.708      0.839      0.816      0.909      0.889      0.930      0.934
     -  9  0.689      0.797      0.789      0.861      0.845      0.881      0.871      0.900
     - 10  0.712      0.794      0.787      0.865      0.847      0.871      0.844      0.849      0.955
     - 11  0.774      0.864      0.757      0.787      0.776      0.809      0.830      0.846      0.939      0.910
     - 12  0.790      0.805      0.794      0.783      0.759      0.797      0.808      0.802      0.858      0.842      0.893
     - 13  0.742      0.766      0.760      0.727      0.708      0.746      0.758      0.748      0.832      0.785      0.861      0.979
     - 14  0.690      0.717      0.741      0.708      0.688      0.732      0.740      0.739      0.825      0.750      0.825      0.953
     - 15  0.675      0.699      0.712      0.684      0.661      0.700      0.706      0.714      0.816      0.737      0.822      0.933
     - 16  0.701      0.727      0.767      0.697      0.675      0.726      0.739      0.694      0.780      0.728      0.804      0.931
     - 17  0.633      0.765      0.829      0.738      0.735      0.741      0.763      0.692      0.756      0.678      0.744      0.822
     - 18 -0.229     -0.243     -0.239     -0.191     -0.232     -0.255     -0.303     -0.189     -0.101     -0.074     -0.156     -0.058
     - 19 -0.127     -0.136     -0.201     -0.215     -0.241     -0.291     -0.302     -0.249     -0.170     -0.081     -0.104     -0.023
     - 20  0.023      0.107     -0.007      0.020      0.065      0.067      0.098      0.192      0.030      0.033      0.078     -0.008
     - 21 -0.150     -0.071     -0.150     -0.148     -0.099     -0.099     -0.077      0.006     -0.130     -0.120     -0.060     -0.172


             13         14         15         16         17         18         19         20

     - 14  0.988
     - 15  0.976      0.992
     - 16  0.961      0.972      0.972
     - 17  0.864      0.873      0.864      0.851
     - 18 -0.048     -0.023      0.023     -0.014     -0.171
     - 19 -0.041     -0.077     -0.041     -0.038     -0.182      0.837
     - 20  0.007     -0.018     -0.040     -0.056      0.091     -0.523     -0.404
     - 21 -0.183     -0.198     -0.207     -0.209     -0.077      0.404     -0.255      0.837
```

continued

```
                                   DEMONSTRATION OF SPRHO

                            CORRESPONDING VALUES OF STUDENT'S

                                      28  DEGREES OF FREEDOM

                            NUMBER OF OBSERVATIONS IS       30

              1          2          3          4          5          6          7          8          9         10         11
               ONE        TWO      THREE       FOUR
    TWO-  2   9.136
  THREE-  3   6.044     11.316
   FOUR-  4   5.718      9.696     16.396
       -  5   5.484      9.826     16.059     39.226
       -  6   5.812      9.246     14.947     27.278     21.433
       -  7   6.648     11.349     16.095     20.095     17.661     34.924
       -  8   5.305      8.164      7.470     11.515     10.262     13.385     13.819
       -  9   5.028      6.992      6.794      8.976      8.351      9.852      9.402     10.897
       - 10   5.365      6.923      6.753      9.123      8.445      9.364      8.340      8.510     16.95
       - 11   6.467      9.096      6.126      6.750      6.519      7.271      7.880      8.389     14.487     11.624
         12   6.807      7.184      6.906      6.657      6.167      6.981      7.262      7.098      8.851      8.247     10.488
         13   5.849      6.307      6.187      5.604      5.310      5.923      6.149      5.955      7.924      6.705      8.954     25.644
       - 14   5.041      5.438      5.834      5.308      5.011      5.693      5.829      5.813      7.727      5.995      7.733     16.581
       - 15   4.845      5.173      5.370      4.960      4.664      5.182      5.275      5.390      7.457      5.773      7.645     13.706
       - 16   5.208      5.598      6.325      5.144      4.845      5.581      5.812      5.095      6.589      5.622      7.163     13.498
       - 17   4.327      6.291      7.831      5.780      5.732      5.843      6.249      5.070      5.108      4.881      5.893      7.646
       - 18  -1.246     -1.323     -1.303     -1.027     -1.265     -1.397     -1.680     -1.019     -0.536     -0.393     -0.837     -0.305
       - 19  -0.679     -0.729     -1.087     -1.163     -1.316     -1.612     -1.676     -1.363     -0.913     -0.428     -0.554     -0.123
       - 20   0.122      0.570     -0.038      0.107      0.346      0.353      0.521      1.036      0.161      0.173      0.416     -0.040
         21  -0.801     -0.375     -0.804     -0.790     -0.525     -0.528     -0.408      0.033     -0.693     -0.641     -0.320     -0.923


             13         14         15         16         17         18         19         20

       - 14  34.060
       - 15  23.595     41.763
       - 16  18.424     21.740     21.889
       - 17   9.065      9.477      9.062      8.558
       - 18  -0.255     -0.120      0.124     -0.075     -0.916
       - 19  -0.219     -0.408     -0.213     -0.201     -0.980      8.081
       - 20   0.035     -0.093     -0.215     -0.296      0.483     -3.249     -2.335
       - 21  -0.987     -1.067     -1.121     -1.129     -0.406     -2.335     -1.397      8.091
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 5 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.55 + network overhead

CONTENTS—(STPAC) SPRHO

pages
1 Identification & Abstract
3 User Instructions
5– 7 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Mann–Whitney U Test
CALLING NAME	(STPAC) MNWHT
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

MNWHT computes results for the Mann–Whitney U test and gives corresponding z scores, probabilities of the z scores, and levels of significance. U's are calculated for all possible pairs of variables. Observations per variable may be unequal. A correction is made for ties. U's will be punched on cards at the option of the user.

REFERENCES

Siegre, S., *Nonparametric Statistics for the Behavioral Sciences* (McGraw–Hill Book Co., Inc., New York, 1956), pp. 116–127.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program

---

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential for MNWHT. Begin- and End-Data Cards also are necessary if the data have not been already read for a previous STPAC program. All other STPAC Control Cards are optional. One additional optional Control Card is specific to MNWHT and is described below.

Punch Card (optional)

Columns	*Contents*
1–5	PUNCH

This card should be included only if the user wishes to have the values of U punched on cards in addition to being printed.

Description of Input Data

Each Data Card should have a variable-identification number (in integer format), followed by a data point (in E or F format). Variable-identification numbers need not be successive, but the program's number of a variable is determined by the order in which it is read. All Data Cards for one variable should be together.

Description of Output

The printed output for a MNWHT problem consists of four triangular matrices. All tables are headed by a user-supplied title. All variables are numbered, and both rows and columns will be named according to the Variable-Names Card, if it is included. Variable identification in each table also includes the number of observations for each variable.

The first table consists of the Mann–Whitney U's for all sets of observations. The second table gives corresponding z scores. A z score in this case should be interpreted cautiously if the number of observations for both of the variables involved is less than 20. Special tables are available to determine the associated

continued

probability in that range. The third table gives the probability associated with each z score.

The fourth table gives the level of significance associated with each of the probabilities.

SAMPLE INPUT

```
EXECUTE MNWHT
TITLE MANN-WHITNEY TEST ONE
VARIABLES 6
FORMAT(I1.2F2.0)
BEGIN DATA
1 9
111
115
2 6
217
2 8
216
2 8
215
210
215
210
215
210
214
211
214
211
214
212
213
212
213
212
213
212
378
 .
 .
 .
513
699
670
653
651



END DATA
STOP
```

SAMPLE OUTPUT

```
                                        MANN-WHITNEY TEST ONE

                                        MANN - WHITNEY U TEST

                    1          2          3          4          5


             N      3          23         5          16         4


-  2 N=    23     29.500
-  3 N=     5      0.0        0.0
-  4 N=    16     10.500     64.000      0.0
-  5 N=     4      3.000     20.500      0.0       30.000
-  6 N=     4      0.0        0.0        9.000      0.0        0.0


                                        MANN-WHITNEY TEST ONE

                                     CORRESPONDING Z SCORES

                    1          2          3          4          5


             N      3          23         5          16         4


-  2 N=    23     -0.404
-  3 N=     5     -2.236     -3.462
-  4 N=    16     -1.532     -3.451     -3.338
-  5 N=     4     -1.061     -1.752     -2.449     -0.192
-  6 N=     4     -2.121     -3.153     -0.245     -3.061     -2.309


                                        MANN-WHITNEY TEST ONE

                  CORRESPONDING PROBABILITIES OF Z SCORES( PROB( X GREATER THAN OR EQUAL TO Z) )

                    1          2          3          4          5


             N      3          23         5          16         4


-  2 N=    23      0.343
-  3 N=     5      0.013      0.000
-  4 N=    16      0.063      0.000      0.000
-  5 N=     4      0.144      0.040      0.007      0.424
-  6 N=     4      0.017      0.001      0.403      0.001      0.010


                                        MANN-WHITNEY TEST ONE

                                        LEVELS OF SIGNIFICANCE

                                     BLANK IF     0.050 < P
                                         * IF     0.025 < P < 0.050
                                        ** IF     0.010 < P < 0.025
                                       *** IF     0.001 < P < 0.010
                                      **** IF             P < 0.001

                    1          2          3          4          5


             N      3          23         5          16         4


-  2 N=    23
-  3 N=     5      **         ****
-  4 N=    16                 ****       ****
-  5 N=     4                    *        ***
-  6 N=     4      **         ****                  ***        **
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) MNWHT

pages
1 Identification & Abstract
3– 4 User Instructions
5– 6 I/O
7 Cost—Contents

DESCRIPTIVE TITLE	Kruskal–Wallis One-Way Analysis of Variance
CALLING NAME	(STPAC) KRWAL
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

KRWAL does a Kruskal–Wallis one-way analysis of variance. The levels need not have equal numbers of observations in each. A correction is made for ties.

REFERENCES

Siegel, *Nonparametric Statistics for the Behavioral Sciences*, (McGraw–Hill Book Co., Inc., New York, 1956).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed by using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various STPAC Control Cards discussed in the STPAC writeup, the Execute and Format cards are essential for KRWAL. Begin-Data and End-Data Cards also are necessary if the data have not been already read for a previous STPAC program. All other STPAC Control Cards are optional. No additional Control Cards are necessary for KRWAL.

Description of Input Data

Each Data Card must contain a level (sample) number in integer format and a data value in E or F format. Level numbers may be any integer values. They need not be consecutive but must be distinct. All Data Cards for a single level (sample) must be together. There is no standard format for this program.

Description of Output

Output consists of degrees of freedom, the value of H, and the chi-square probability of H. If there are ties in the data, the number of groups of ties, the total number of ties, and the correction factor also will be printed. If the number of observations per level (sample) is less than five, caution should be exercised in the interpretation of the chi-square probability. Tables are available for probabilities of H in this lower range.

SAMPLE INPUT

```
EXECUTE KRWAL
TITLE SECOND TEST OF KRUSKAL WALLIS ANOVA-FIRST DATA RANKED
VARIABLES 8
FORMAT(I1,F3.1)
BEGIN DATA
1 85
1275
1475
141
156
1545
1 6
1475
1275
1 1
2525
2275
241
2525
214
2155
2 85
2 5
3475
  .
  .
  .
6185
723
7125
7125
7185
7 25
7 25
8185
8155
835



8 4
END DATA
STOP
```

SAMPLE OUTPUT

```
        KRUSKALL-WALLIS ONE WAY ANALYSIS OF VARIANCE BY RANKS
     SECOND TEST OF KRUSKAL WALLIS ANOVA-FIRST DATA RANKED
                    7   DEGREES OF FREEDOM
                    H =    18.565
13 GROUPS OF TIES.    56 TOTAL TIES.   CORRECTION FACTOR IS 0.9945318
               CHI SQUARE PROBABILITY =  0.009664
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) KRWAL

pages

1 Identification & Abstract
3 User Instructions
5– 6 I/O
7 Cost—Contents

DESCRIPTIVE TITLE	Analysis-of-Variance Method of Unweighted Means
CALLING NAME	(STPAC) ANOVUM
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

ANOVUM is designed to handle a factorial analysis of variance with either equal or unequal numbers of observations in subclasses. The method of unweighted means is based on the assumption that the population from which the sample is drawn has proportional or equal subclass numbers. It can be relied upon only if the subclass numbers are approximately equal and presumably represent a population with equal numbers. The analysis is performed upon the means for each cell, and each sum of squares is multiplied by the harmonic mean of the number of observations per cell. Up to 150 variables (separate analyses) will be permitted with data code = 1. Only 1 variable is acceptable with data code = 0.

A Bartlett's test for homogeneity of variance is performed on the raw data. If the chi square computed in this test is significant at the 0.05 level, the conclusions of the analysis of variance should be interpreted cautiously. If cell frequencies are greater than 1, and if they are equal, a Scheffe's test will be performed on main effects.

REFERENCES

Winer, B.J., *Statistical Principles in Experimental Design* (McGraw-Hill Book Co., Inc., New York, 1962), pp. 95–96, 222–224.

USER INSTRUCTIONS

This Program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute and Format Cards are essential for ANOVUM. Begin- and End-Data Cards are also necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional. Some additional Control Cards are required and some optional Control Cards are specific to ANOVUM (see below).

Treatments Card (required)

Columns	*Contents*
1–10	TREATMENTS
12	N, the number of treatments
14–19	LEVELS
21–80	$L_1,L_2,\cdots,L_N$; L_i is the number of levels for the i^{th} treatment

Data-Code Card (required)

Columns	*Contents*
1– 9	DATA CODE
11	1: there is one data point per variable per card, preceded by cell identification 0: all data for a single cell are punched on a single card or card sequence. If more than one Data Card per cell is required (i.e., if cell identification and data use more than 80 columns), a sequence number must be included on each card, following the cell identification. See Data Cards, below.

Equal or Unequal Card (required only if data code is 0)

Columns	*Contents*
1–80	EQUAL n SEQUENCED m PERCARD or

continued

Columns	*Contents*
	UNEQUAL n SEQUENCED m PERCARD

This card indicates whether the cell frequencies are equal or unequal. n must be the number of observations per cell if cells are equal or the maximum number of observations per cell if cells are unequal.

The next three fields may be omitted if only one Data Card per cell is necessary. If, however, more than one Data Card per cell is necessary, the last three fields of this card must be included. m, in this case, is the number of data points per card.

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card should be included only if the user desires punched-card output of means.

Cell Summary Card (optional)

Columns	*Contents*
1–12	CELL SUMMARY

This card calls for a summary table of cell frequencies, means, variance, and degrees of freedom to be printed.

Treatment Summary Card (optional)

Columns	*Contents*
1–17	TREATMENT SUMMARY

This card calls for a summary table of totals, number of observations, means, and sums of squares for each treatment group and combination to be printed.

Missing-Data Card (optional)

Columns	*Contents*
1–12	MISSING DATA
14–21	x, any number up to eight digits, including a decimal, which will be read as a code for missing data

Include this card only if some data are missing. If this card is omitted, it will be assumed that all fields contain data (blanks

continued

will be read as zero). If this card is included with Cols. 14–21 blank, zeros and blanks in the Data Cards will be treated as missing data. If Cols. 14–21 are punched, the number x in a data field will be treated as a case of missing data.

Scheffe Card (optional)

Columns	*Contents*
1– 7	SCHEFFE
9–80	Desired level of alpha for the Scheffe test. (The default level of alpha is 0.05.)

No standard format is provided for this program; format must be supplied by the user as specified in the requirements for STPAC.

Data Cards

Data Cards are punched with code numbers signifying level for each treatment, starting at the lefthand side of the card (cell identification). These must be in integer format. The first (I_1) indicates level for treatment 1; the second (I_2) indicates level for treatment 2; etc. If any of the $I_1,\cdots,I_N$ are blank or zero, the card will be ignored by the program. If any of the $I_1,\cdots,I_N$ are greater than the maximum number of levels specified on the Treatments Card, the analysis will not be performed.

If the data code is 1, a single data point for each of the variables specified (in F, D, or E format) follows the cell identification. Data Cards with this data code may be in any order.

If the data code is 0, and cell frequencies are unequal, the number of data points in that particular cell (in I format) must immediately follow the cell identification. If a sequence number is also required, it (also in I format) follows the number of data points. In the case where more than one card per cell is needed under data code = 0, cell identification, number of data points, and sequence number must be the first items of information on *every* card. Data fields follow the preceding fields. All cards for a single cell must be together.

Data Code = 1 FORM: $I_1,\cdots,I_N\ X_1,\cdots,X_K$

where $I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; $X_1,\cdots,X_K$ are data points, one for each of K specified variables.

continued

Data Code = 0

FORM 1: $I_1,\cdots,I_N\ X_1,\cdots,X_n$
WHEN: cell frequencies are equal
all data for a cell on single card

$I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; $X_1,\cdots,X_n$ are the n data points as specified on the Equal Card.

FORM 2: $I_1,\cdots,I_N\ M\ X_1,\cdots,X_M$
WHEN: cell frequencies are *un*equal
data for a cell on single card

$I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (an integer) is the number of data points in that particular cell; $X_1,\cdots X_M$ are the M data points.

FORM 3: $I_1,\cdots,I_N\ J\ X_1,\cdots,X_m$
WHEN: cell frequencies are equal
more than 1 card necessary for all data points in each cell

$I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; J (integer) is the card sequence number; $X_1,\cdots,X_m$ are the m data points per sequenced card, as specified on the Equal Card.

FORM 4: $I_1,\cdots,I_N\ M\ J\ X_1,\cdots X_m$
WHEN: cell frequencies are *un*equal
more than 1 card necessary for all data points in each cell

$I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (integer) is the number of data points in that particular cell and must be repeated for every card in that cell; J (integer) is the card sequence number; $X_1,\cdots,X_m$ are the m data points per sequenced card, as specified on the Unequal Card.

Description of Output

For each cell, mean, number, variance, degrees of freedom, and cell identification are printed at the option of the user. Chi square, correction factor, degrees of freedom, and probability are printed for the Bartlett's test for homogeneity of variance. For each treatment group and combination, total, number of observations, mean, and sum of squares are given at the option of the user. An analysis-of-

continued

variance summary table, including treatment names (according to Treatment-Names Card), sums of squares, degrees of freedom, mean squares, and F ratios, and their probabilities will be given, along with the output from the Scheffe's test if applicable.

REFERENCES

Winer, B.J., *Statistical Principles in Experimental Design* (McGraw-Hill Book Co., Inc., New York, 1962), pp. 95–96, 222–224.

SAMPLE INPUT

```
EXECUTE ANOVUM
TITLE DEMONSTRATION OF ANOVUM WITH EQUAL OBSERVATIONS PER CELL
TREATMENTS 3 LEVELS 2 2 2
TREATMENT SUMMARY
VARIABLES 1
CELL SUMMARY
TREATMENT NAMES 1=SEX,2=INCOME,3=JOB
FORMAT(3I1,2F2.0)
DATA CODE 0
EQUAL 2
BEGIN DATA
1112536
1122943
1211513
1221012
211 9 8
21211 7
221 5 4
222 6 8
END DATA
STOP
```

SAMPLE OUTPUT

```
                                   DEMONSTRATION OF ANOVUM WITH EQUAL OBSERVATIONS PER CELL

                                                  VARIABLE NUMBER      1

   MEAN        N       VARIANCE      DEGREES OF FREEDOM       CELL
 30.500        2       60.5000               1                   1    1    1
 36.000        2       98.0000               1                   1    1    2
 14.000        2        2.0000               1                   1    2    1
 11.000        2        2.0000               1                   1    2    2
  8.500        2        0.5000               1                   2    1    1
  9.000        2        8.0000               1                   2    1    2
  4.500        2        0.5000               1                   2    2    1
  7.000        2        2.0000               1                   2    2    2

           CORRECTION FACTOR IS        1.37500

 BARTLETT'S TEST OF HOMOGENEITY OF VARIANCE:
CHI SQUARE =       9.5680  PROBABILITY = 0.214402
          7 DEGREES OF FREEDOM
 USE CAUTION IN INTERPRETATION IF ANY CELL FREQUENCIES ARE LESS THAN 3
```

continued

000 0051(p)

```
                                   DEMONSTRATION OF ANOVUM WITH EQUAL OBSERVATIONS PER CELL
CELL FREQUENCIES ARE EQUAL - STANDARD ANALYSIS FOLLOWS
                                                                VARIABLE NUMBER      1
TREATMENT LEVEL                 TOTAL         N         MEAN       SUM OF SQUARES

 1
          1                  183.0000         8      22.8750        0.418613D 04
          2                   58.0000         8       7.2500        0.420500D 03

 2
          1                  168.0000         8      21.0000        0.352800D 04
          2                   73.0000         8       9.1250        0.666125D 03

 3
          1                  115.0000         8      14.3750        0.165313D 0
          2                  126.0000         8      15.7500        0.198450D 0

 12
          1    1             133.0000         4      33.2500        0.442225D 04
          2    1              35.0000         4       8.7500        0.306250D 03
          1    2              50.0000         4      12.5000        0.625000D 03
          2    2              23.0000         4       5.7500        0.132250D 03

 13
          1    1              89.0000         4      22.2500        0.198025D 04
          2    1              26.0000         4       6.5000        0.169000D 03
          1    2              94.0000         4      23.5000        0.220900D 04
          2    2              32.0000         4       8.0000        0.256000D 03

 23
          1    1              78.0000         4      19.5000        0.152100D 04
          2    1              37.0000         4       9.2500        0.342250D 03
          1    2              90.0000         4      22.5000        0.202500D 04
          2    2              36.0000         4       9.0000        0.324000D 03

 123
          1    1    1         61.0000         2      30.5000        0.186050D 04
          2    1    1         17.0000         2       8.5000        0.144500D 03
          1    2    1         28.0000         2      14.0000        0.392000D 03
          2    2    1          9.0000         2       4.5000        0.405000D 02
          1    1    2         72.0000         2      36.0000        0.259200D 04
          2    1    2         18.0000         2       9.0000        0.162000D 03
          1    2    2         22.0000         2      11.0000        0.242000D 03
          2    2    2         14.0000         2       7.0000        0.980000D 02
```

continued

000 0051(p)

000 0051(p)

000 0051(p)

```
                                        DEMONSTRATION OF ANOVUM WITH EQUAL OBSERVATIONS PER CELL

                                                              VARIABLE NUMBER      1

                  ANALYSIS OF VARIANCE SUMMARY TABLE

  SOURCE            SUMS OF SQUARES       DF         MEAN SQUARES         F RATIO     PROBABILITY
     1       SEX     0.9765630 03          1         0.9765630 03          45.029         0.0000
     2    INCOME     0.5640630 03          1         0.5640630 03          26.009         0.0000
     3       JOB     0.7562500 01          1         0.7562500 01           0.349         0.5711
    12               0.3150630 03          1         0.3150630 03          14.527         0.0052
    13               0.6250000-01          1         0.6250000-01           0.003         0.9585
    23               0.1056250 02          1         0.1056250 02           0.487         0.5050
   123               0.2756250 02          1         0.2756250 02           1.271         0.2923

   ERROR             0.1735000 03          8         0.2168750 02

CORRECTED TOTAL SUM OF SQUARES =  0.207494D 04


UNCORRECTED TOTAL SUM OF SQUARES =  0.570500D 04

                TREATMENT  LEVEL  DIFFERS  LEVEL  WITH ALPHA = 0.0500.  DIFFFRENCE  SCHEFFE'S  ESTIMATE OF VARIANCE
                                   FROM                                   BETWEEN    CONSTANT     OF DIFFERENCES
                                                                           MEANS

                    1                                                                  2.316        0.2328D 01
                             1               2                          0.1563D 02

                    2                                                                  2.316        0.2328D 01
                             1               2                          0.1188D 02
```

000 0051(p)

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 8 seconds on the central-processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.88 + network overhead

CONTENTS—(STPAC) ANOVUM

pages	
1	Identification & Abstract
3- 7	User Instructions
9—12	I/O
13	Cost—Contents

DESCRIPTIVE TITLE	Analysis of Variance with Unequal Subclass Numbers (Method of Expected Subclass Numbers)
CALLING NAME	(STPAC) ANOVES
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

ANOVES is designed to handle a factorial analysis of variance with unequal numbers of observations in subclasses (cells). A problem with equal subclass numbers is acceptable, although it is not advisable to use ANOVES for a balanced design with equal subclass numbers owing to its relatively long computing time as compared to that of ANOVUM. Up to 150 variables (separate analyses) will be permitted with data code = 1. Only one variable is acceptable with data code = 0.

This method of analyzing data of multiple classifications with unequal subclass numbers is based on the assumption that the population from which the sample is drawn has proportional or equal subclass numbers. Under its fundamental hypotheses, this method affords an estimate of both the main effects and the interactions. However, every subclass must contain at least one observed value.

If the computed chi-square value for disproportionality is significant, a reduction factor will be calculated and all the sums of squares (except the with-in subclasses, which is calculated from the original data) will be reduced accordingly. This is an attempt to remove part of the disproportionality of the data.

continued

The probability associated with the chi-square value computed by using subclass numbers is calculated and the 0.05 probability point is considered critical in determining whether a reduction factor will be computed or not.

The reduction factor is

$$\frac{N}{\sum_{1}^{K} E^2/A}$$

where N is the total number of observations, K is the number of cell subclasses, E is the expected number of observations in a cell, and A is the actual number of observations in the same cell.

A Bartlett's test for homogeneity of variance is performed on the raw data. If the chi square computed in this test is significant, the conclusions of the analysis of variance should be interpreted cautiously.

REFERENCES

Snedecor, G.W., and Cox, G.M., "Disproportionate Subclass Numbers in Tables of Multiple Classification," Ia. Agriculture Sta. Res. Bul. *180*, 233–272 (1935).

Bennett, K.R. (personal correspondence to J. Streeter).

Winer, B.J., *Statistical Principles in Experimental Design* (McGraw--Hill Book Co., Inc., New York, 1962).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute and Format Cards are essential for ANOVES. Begin- and End-Data Cards are also necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional. Some additional Control Cards are required and some optional Control Cards are specific to ANOVES (see below).

Treatments Card (required)

Columns	*Contents*
1–10	TREATMENTS
12	N, the number of treatments
14–19	LEVELS
21–80	$L_1, L_2, \cdots, L_N$; L_i is the number of levels for the i^{th} treatment

Data-Code Card (required)

Columns	*Contents*
1– 9	DATA CODE
11	1: there is one data point per variable per card, preceded by cell identification 0: all data for a single cell are punched on a single card or card sequence. If more than one Data Card per cell is required (i.e., if cell identification and data use more than 80 columns), a sequence number must be included on each card, following the cell identification. See Data Cards, below.

Equal or Unequal Card (required only if data code is 0)

Columns	*Contents*
1–80	EQUAL n SEQUENCED m PERCARD or

continued

Columns	*Contents*
	UNEQUAL n SEQUENCED m PERCARD

This card indicates whether the cell frequencies are equal or unequal. n must be the number of observations per cell if cells are equal or the maximum number of observations per cell if cells are unequal.

The next three fields may be omitted if only one Data Card per cell is necessary. If, however, more than one Data Card per cell is necessary, the last three fields of this card must be included. m, in this case, is the number of data points per card.

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card should be included only if the user desires punched-card output of means.

Cell Summary Card (optional)

Columns	*Contents*
1–12	CELL SUMMARY

This card calls for a summary table of cell frequencies, means, variance, and degrees of freedom to be printed.

Treatment Summary Card (optional)

Columns	*Contents*
1–17	TREATMENT SUMMARY

This card calls for a summary table of totals, number of observations, means, and sums of squares for each treatment group and combination to be printed.

Missing-Data Card (optional)

Columns	*Contents*
1–12	MISSING DATA
14–21	x, any number up to eight digits, including a decimal, which will be read as a code for missing data

Include this card only if some data are missing. If this card is omitted, it will be assumed that all fields contain data (blanks

continued

will be read as zero). If this card is included with Cols. 14–21 blank, zeros and blanks in the Data Cards will be treated as missing data. If Cols. 14–21 are punched, the number x in a data field will be treated as a case of missing data.

No standard format is provided for this program; format must be supplied by the user as specified in the requirements for STPAC.

Data Cards

Data Cards are punched with code numbers signifying level for each treatment, starting at the lefthand side of the card (cell identification). These must be in integer format. The first (I_1) indicates level for treatment 1; the second (I_2) indicates level for treatment 2; etc. If any of the $I_1,\cdots,I_N$ are blank or zero, the card will be ignored by the program. If any of the $I_1,\cdots,I_N$ are greater than the maximum number of levels specified on the Treatments Card, the analysis will not be performed.

If the data code is 1, a single data point for each of the variables specified (in F, D, or E format) follows the cell identification. Data Cards with this data code may be in any order.

If the data code is 0, and cell frequencies are unequal, the number of data points in that particular cell (in I format) must immediately follow the cell identification. If a sequence number is also required, it (also in I format) follows the number of data points. In the case where more than one card per cell is needed under data code = 0, cell identification, number of data points, and sequence number must be the first items of information on *every* card. Data fields follow the preceding fields. All cards for a single cell must be together.

Data Code = 1 FORM: $I_1,\cdots,I_N\ X_1,\cdots,X_K$

where $I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; $X_1,\cdots,X_K$ are data points, one for each of K specified variables.

Data Code = 0 FORM 1: $I_1,\cdots,I_N\ X_1,\cdots,X_n$
WHEN: cell frequencies are equal
all data for a cell on single card

$I_1,\cdots,I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; $X_1,\cdots,X_n$ are the n data points as specified on the Equal Card.

continued

FORM 2: $I_1, \cdots, I_N$ M $X_1, \cdots, X_M$
WHEN: cell frequencies are *un*equal
data for a cell on single card

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (an integer) is the number of data points in that particular cell; $X_1, \cdots, X_M$ are the M data points.

FORM 3: $I_1, \cdots, I_N$ J $X_1, \cdots, X_m$
WHEN: cell frequencies are equal
more than 1 card necessary for all data points in each cell

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; J (integer) is the card sequence number; $X_1, \cdots, X_m$ are the m data points per sequenced card, as specified on the Equal Card.

FORM 4: $I_1, \cdots, I_N$ M J $X_1, \cdots, X_m$
WHEN: cell frequencies are *un*equal
more than 1 card necessary for all data points in each cell

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (integer) is the number of data points in that particular cell and must be repeated for every card in that cell; J (integer) is the card sequence number; $X_1, \cdots, X_m$ are the m data points per sequenced card, as specified on the Unequal Card.

Description of Output

The chi-square value of the Bartlett's test for homogeneity of variance is printed out, along with its associated probability. For each cell, the total, number of observations, mean, expected total, and expected mean are printed also. The chi-square value for disproportionality is printed along with its associated probability. If this chi square is significant at the 0.05 level, the reduction factor is also printed. For each treatment group and combination, expected total, expected number of observations, and sum of squares are given. An analysis-of-variance summary table, including treatment names (according to Treatment-Names Card), sums of squares, degrees of freedom, mean squares, and F ratios, is printed last. Card output of means is provided at the option of the user.

SAMPLE INPUT

```
EXEC ANOVES
TREATMENTS 3 LEVELS 2 2 2
DATA CODE 1
TREATMENT NAMES TRANSFER
FORMAT(3I1,F2.0)
TREATMENT SUMMARY
VARIABLES 1
BEGIN DATA
11125
11229
12115
12210
211 9
21211
221 5
222 6
11131
11226
22115
22213
END DATA
STOP
```

SAMPLE OUTPUT

```
                                     BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCE

                                                    VARIABLE NUMBER      1

BARTLETT'S TEST CANNOT BE COMPUTED BECAUSE ONE OR MORE CELL FREQUENCIES EQUAL 1;
 OR BECAUSE ONE OR MORE VARIANCES EQUAL 0.


                                                    VARIABLE NUMBER      1

CHI SQUARE OF DISPROPORTIONALITY OF SUBCLASS FREQUENCIES =        1.3333 P = 0.248213


                                                    VARIABLE NUMBER      1

                                              TREATMENT SUMMARY - MARGINAL VALUES

FACTOR    LEVEL                  EXPECTED TOTAL   EXPECTED NO.      EXPECTED MEAN        SUM OF SQUARES

 1
          1                            120.7500          6.000            20.1250          0.2430090 04
          2                             59.2500          6.000             9.8750          0.5850940 03

 2
          1                            113.2500          6.000            18.8750          0.2137590 04
          2                             66.7500          6.000            11.1250          0.7425940 03

 3
          1                             93.0000          6.000            15.5000          0.1441500 04
          2                             87.0000          6.000            14.5000          0.1261500 04

 12
          1    1                        83.2500          3.000            27.7500          0.2310190 04
          2    1                        30.0000          3.000            10.0000          0.3000000 03
          1    2                        37.5000          3.000            12.5000          0.4687500 03
          2    2                        29.2500          3.000             9.7500          0.2851880 03

 13
          1    1                        64.5000          3.000            21.5000          0.1386750 04
          2    1                        28.5000          3.000             9.5000          0.2707500 03
          1    2                        56.2500          3.000            18.7500          0.1054690 04
          2    2                        30.7500          3.000            10.2500          0.3151880 03

 23
          1    1                        55.5000          3.000            18.5000          0.1026750 04
          2    1                        37.5000          3.000            12.5000          0.4687500 03
          1    2                        57.7500          3.000            19.2500          0.1111690 04
          2    2                        29.2500          3.000             9.7500          0.2851880 03

 123
          1    1    1                   42.0000          1.500            28.0000          0.1176000 04
          2    1    1                   13.5000          1.500             9.0000          0.1215000 03
          1    2    1                   22.5000          1.500            15.0000          0.3375000 03
          2    2    1                   15.0000          1.500            10.0000          0.1500000 03
          1    1    2                   41.2500          1.500            27.5000          0.1134380 04
          2    1    2                   16.5000          1.500            11.0000          0.1815000 03
          1    2    2                   15.0000          1.500            10.0000          0.1500000 03
          2    2    2                   14.2500          1.500             9.5000          0.1353750 03
                                                     VARIABLE NUMBER      1

                    ANALYSIS OF VARIANCE SUMMARY TABLE

   SOURCE             SUMS OF SQUARES       DF       MEAN SQUARES        F RATIO
        1               0.3151880 03         1        0.3151880 03        12.997
        2               0.1801880 03         1        0.1801880 03         7.430
        3               0.3000000 01         1        0.3000000 01         0.124
       12               0.1687500 03         1        0.1687500 03         6.959
       13               0.9187500 01         1        0.9187500 01         0.379
       23               0.9187500 01         1        0.9187500 01         0.379
      123               0.7500000 00         1        0.7500000 00         0.031

    ERROR               0.9700000 02         4        0.2425000 02


UNCORRECTED TOTAL SUM OF SQUARES =  0.4085000 04
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) ANOVES

pages

1– 2 Identification & Abstract
3– 6 User Instructions
7– 8 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Analysis of Variance with Repeated Measures (Proportionality Assumed)
CALLING NAME	(STPAC) AOVRM
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

AOVRM is designed to handle a factorial analysis of variance with repeated measures and either equal or unequal numbers of observations in subclasses (cells). A problem with no repeated measures is acceptable, although it is not advisable to use AVORM for such a design owing to the relatively long computing time as compared with that of ANOVM or ANOVES.

This method of analyzing data of multiple classifications with unequal subclass numbers is based on the assumption that the population from which the sample is drawn has proportional or equal subclass numbers. This method allows calculation of all main effects and interactions. However, every subclass must contain at least one subject and the data for every subject included must be complete.

REFERENCES

Winer, B.J., *Statistical Principles in Experimental Design* (McGraw-Hill Book Co., Inc., New York, 1962).

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute, Variables, and Format Cards are essential for AVORM. Begin- and End-Data Cards are also necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional. Some additional Control Cards are required and some optional Control Cards are specific to AOVRM (see below).

Treatments Card (required)

Columns	*Contents*
1–10	TREATMENTS
12	N, the number of treatments
14–19	LEVELS
21–80	$L_1, L_2, \cdots, L_N$; L_i is the number of levels for the i^{th} treatment

This card form is used *only* if *no* repeated measures are involved in the design. For repeated treatments, use the following form.

Columns	*Contents*
1–10	TREATMENTS
12	N, the total number of treatments
14–21	REPEATED
23	M, the number of repeated treatments
25–30	LEVELS
32–80	$L_1, L_2, \cdots, L_n$, as above. The repeated-measure treatments must be the last M of the N treatments

Data-Code Card (required)

Columns	*Contents*
1– 9	DATA CODE

continued

Data-Code Card

Columns	*Contents*
11	1: there is one data point per card, preceded by cell identification for all treatements, and the subject number 0: all data for a single subject are punched on a single card or card sequence. If more than one Data Card per subject is required (i.e., if cell identification and data use more than 80 columns), the format supplied by the user must take this into account. The program will not. Cell identification in this case is only for the nonrepeated measures and the subject number

Observations Card (required)

Columns	*Contents*
1–12	OBSERVATIONS
14–80	N, an integer, the number of observations per cell (in the case of equal cell frequencies) or the maximum number of subjects per cell (in the case of unequal cell frequencies)

Punch Card (optional)

Columns	*Contents*
1– 5	PUNCH

This card should be included only if the user desires punched-card output of means.

Cell Summary Card (optional)

Columns	*Contents*
1–12	CELL SUMMARY

This card calls for a summary table of cell frequencies, means, variance, and degrees of freedom to be printed.

Treatment Summary Card (optional)

Columns	*Contents*
1–17	TREATMENT SUMMARY

This card calls for a summary table of totals, number of observations, means, and sums of squares for each treatment group and combination to be printed.

continued

Missing-Data Card (required if there are unequal number of subjects in subclasses)

Columns	*Contents*
1–12	MISSING DATA
14–21	x, any number up to eight digits, including a decimal, which will be read as a code for missing data

If this card is omitted, it will be assumed that all fields contain data (blanks will be read as zero). If this card is included with Cols. 14–21 blank, zeros and blanks in the Data Cards will be treated as missing data. If Cols. 14–21 are punched, the number x in a data field will be treated as a case of missing data.

Data Cards

If data code = 1, Data Cards are punched with code numbers signifying level for each treatment, the code for the subject (all in integer format), followed by a single data point (in F, E, or D format).

If data code = 0, Data Cards are punched with code numbers signifying level for each nonrepeated treatment, the code for the subject (all in integer format), followed by *all* the data for that subject (in F, E, or D format).

The order in which the data are punched if data code = 0 is critical, since order is the only way in which the program can identify the data. The first treatment level to be changed should be the last repeated-measure treatment. The last to be changed should be the first repeated-measure treatment; i.e., data for the levels of the innermost treatment in the nest of treatments should be punched first, and so on, to the outermost.

Subject codes must be consecutive, beginning with 1, for each individual cell, whether data code = 0 or 1.

Description of Output

For each cell, the mean, number, variance, degrees of freedom, and cell identification are printed at the option of the user. For each treatment group and combination, the total, number of observations, mean, and sum of squares are given, also at the user's option.

An analysis-of-variance summary table—including treatment names (according to the Treatment-Names Card, if provided), sums of squares, degrees of freedom, mean squares, and F ratios for all treatments, interactions, and error terms—is printed last.

Punched-card output of cell means is provided optionally.

SAMPLE INPUT

```
EXECUTE AOVRM
TREATMENTS 4 REPEATED 2 LEVELS 2 2 3 2
DATA CODE 0
OBSERVATIONS 2
TREATMENT SUMMARY
VARIABLES 1
FORMAT(3I1,6F2.0)
TREATMENT NAMES 1=SEX,2=OCCUPATION,3=PERIOD,4=TEST
BEGIN DATA
111415239463847
112365341493645
121425043503548
122435445534049
211354938473747
212335039463248
221385134503147
222415240463545
END DATA
STOP
```

SAMPLE OUTPUT

```
TREATMENT  LEVEL                         TOTAL      N        MEAN      SUM OF SQUARES

 1
             1                        1075.0000    24      44.7917       0.481510D 05
             2                        1011.0000    24      42.1250       0.425884D 05

 2
             1                        1024.0000    24      42.6667       0.436907D 05
             2                        1062.0000    24      44.2500       0.469935D 05

 3
             1                         720.0000    16      45.0000       0.324000D 05
             2                         706.0000    16      44.1250       0.311523D 05
             3                         660.0000    16      41.2500       0.272250D 05

 4
             1                         912.0000    24      38.0000       0.346560D 05
             2                        1174.0000    24      48.9167       0.574282D 05

 12
             1    1                    523.0000    12      43.5833       0.227941D 05
             2    1                    501.0000    12      41.7500       0.209148D 05
             1    2                    552.0000    12      46.0000       0.253920D 05
             2    2                    510.0000    12      42.5000       0.216750D 05

                          .                                          .
                          .                                          .
                          .                                          .

 134
             1    1    1               162.0000     4      40.5000       0.656100D 04
             2    1    1               147.0000     4      36.7500       0.540225D 04
             1    2    1               168.0000     4      42.0000       0.705600D 04
             2    2    1               151.0000     4      37.7500       0.570025D 04
             1    3    1               149.0000     4      37.2500       0.555025D 04
             2    3    1               135.0000     4      33.7500       0.455625D 04
             1    1    2               209.0000     4      52.2500       0.109203D 05
             2    1    2               202.0000     4      50.5000       0.102010D 05
             1    2    2               199.0000     4      49.5000       0.980100D 04
             2    2    2               189.0000     4      47.2500       0.893025D 04
             1    3    2               189.0000     4      47.2500       0.893025D 04
             2    3    2               187.0000     4      46.7500       0.874225D 04

 234
             1    1    1               145.0000     4      36.2500       0.525625D 04
             2    1    1               164.0000     4      41.0000       0.672400D 04
             1    2    1               157.0000     4      39.2500       0.616225D 04
             2    2    1               162.0000     4      40.5000       0.656100D 04
             1    3    1               143.0000     4      35.7500       0.511225D 04
             2    3    1               141.0000     4      35.2500       0.497025D 04
             1    1    2               204.0000     4      51.0000       0.104040D 05
             2    1    2               207.0000     4      51.7500       0.107123D 05
             1    2    2               188.0000     4      47.0000       0.883600D 04
             2    2    2               199.0000     4      49.7500       0.990025D 04
             1    3    2               187.0000     4      46.7500       0.874225D 04
             2    3    2               189.0000     4      47.2500       0.893025D 04

 1234
             1    1    1    1           77.0000     2      38.5000       0.296450D 04
             2    1    1    1           68.0000     2      34.0000       0.231200D 04
             1    2    1    1           85.0000     2      42.5000       0.361250D 04
```

continued

```
   2   2   1   1            79.0000     2      39.5000        0.312050D 04
   1   1   2   1            80.0000     2      40.0000        0.320000D 04
   2   1   2   1            77.0000     2      38.5000        0.296450D 04
   1   2   2   1            88.0000     2      44.0000        0.387200D 04
   2   2   2   1            74.0000     2      37.0000        0.273800D 04
   1   1   3   1            74.0000     2      37.0000        0.273800D 04
   2   1   3   1            69.0000     2      34.5000        0.238050D 04
   1   2   3   1            75.0000     2      37.5000        0.281250D 04
   2   2   3   1            66.0000     2      33.0000        0.217800D 04
   1   1   1   2           105.0000     2      52.5000        0.551250D 04
   2   1   1   2            99.0000     2      49.5000        0.490050D 04
   1   2   1   2           104.0000     2      52.0000        0.540800D 04
   2   2   1   2           103.0000     2      51.5000        0.530450D 04
   1   1   2   2            95.0000     2      47.5000        0.451250D 04
   2   1   2   2            93.0000     2      46.5000        0.432450D 04
   1   2   2   2           103.0000     2      51.5000        0.530450D 04
   2   2   2   2            96.0000     2      48.0000        0.460800D 04
   1   1   3   2            92.0000     2      46.0000        0.423200D 04
   2   1   3   2            95.0000     2      47.5000        0.451250D 04
   1   2   3   2            97.0000     2      48.5000        0.470450D 04
   2   2   3   2            92.0000     2      46.0000        0.423200D 04


               ANALYSIS OF VARIANCE SUMMARY TABLE

  SOURCE           SUMS OF SQUARES      DF      MEAN SQUARES       F RATIO

BETWEEN SUBJECTS

  1         SEX    0.853333D 02          1      0.853333D 02        11.571
  2    OCCUPATI    0.300833D 02          1      0.300833D 02         4.079
  12               0.833333D 01          1      0.833333D 01         1.130
   ERROR           0.295000D 02          4      0.737500D 01


WITHIN SUBJECTS

  3      PERIOD    0.123167D 03          2      0.615833D 02        29.859
  13               0.316667D 01          2      0.158333D 01         0.768
  23               0.161667D 02          2      0.808333D 01         3.919
  123              0.206667D 02          2      0.103333D 02         5.010
   ERROR           0.165000D 02          8      0.206250D 01


  4        TEST    0.143008D 04          1      0.143008D 04       162.664
  14               0.163333D 02          1      0.163333D 02         1.858
  24               0.750000D 00          1      0.750000D 00         0.085
  124              0.333333D 00          1      0.333333D 00         0.038
   ERROR           0.351667D 02          4      0.879167D 01


  34               0.381667D 02          2      0.190833D 02         5.485
  134              0.666667D 00          2      0.333333D 00         0.096
  234              0.185000D 02          2      0.925000D 01         2.659
  1234             0.316667D 01          2      0.158333D 01         0.455
   ERROR           0.278333D 02          8      0.347917D 01


GRAND TOTAL =  0.208600D 04
NUMBER OF DATA POINTS =         48
CORRECTION FACTOR =  0.906541D 05
UNCORRECTED SUM OF SQUARES =  0.925580D 05
```

continued

```
ACCOUNT   DATE        NAME OF PERSON          USER      MAXIMUM    ACTUAL    MAXIMUM    ACTUAL    COST        JOB NAME
NUMBER    PROCESSED   REQUESTING RUN          IDENT.    TIME       TIME      RECORDS    RECORDS           SITE CAT. SERIAL

S4002     12/08/69    DAUBERT N C             AOVRM        100         9       2000        342    $.00     AA   3   38050
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 5 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.55 + network overhead

CONTENTS—(STPAC) AOVRM

pages
1 Identification & Abstract
3– 5 User Instructions
7–10 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Bartlett's Test for Homogeneity of Variance
CALLING NAME	(STPAC) BARTL
INSTALLATION NAME	The Pennsylvania State University Computation Center
AUTHOR(S) AND AFFILIATION(S)	Nancy C. Daubert The Pennsylvania State University Computation Center
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/67
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	William H. Verity, 104 Computer Building, The Pennsylvania State University, University Park, Pa. 16802 Tel.: (814) 865-9527

FUNCTIONAL ABSTRACT

BARTL performs a Bartlett's test for homogeneity of variance on up to 6000 subgroups (or cells). Subgroups may be equal or unequal.

REFERENCES

Winer, B.J. *Statistical Principles in Experimental Design* (McGraw-Hill Book Co., Inc., New York, 1962), pp. 95–96.

USER INSTRUCTIONS

This program is one of a collection of programs that are all executed using the library program STPAC. The writeup for STPAC must be consulted for the description of additional Control Cards and procedures for running this program.

Control Cards and Their Functions

Of the various Control Cards discussed in the STPAC writeup, the Execute and Format Cards are essential for BARTL. Begin- and End-Data Cards are also necessary if the data have not already been read for a previous STPAC program. All other STPAC Control Cards are optional. Some additional Control Cards are required and some optional Control Cards are specific to BARTL (see below).

Treatments Card (required only if data code = 0 or 1—i.e., data are punched as for analysis of variance)

Columns	*Contents*
1–10	TREATMENTS
12	N, the number of treatments
14–19	LEVELS
21–80	$L_1, L_2, \cdots, L_N$; L_i is the number of levels for the i^{th} treatment

Data-Code Card (required)

Columns	*Contents*
1– 9	DATA CODE
11	1 or 2: there is one data point per card, preceded by subgroup identification. Further differentiation between these two codes will be included in the section on Data Cards. 0: all data for a single subgroup are punched on a single card or card sequence. If more than one Data Card per subgroup is required (i.e., if subgroup identification and data use more than 80 columns), a sequence number must be included on each card, following the subgroup identification. See Data Cards, below.

continued

Equal or Unequal Card (required only if data code is 0)

Columns	*Contents*
1–80	EQUAL n SEQUENCED m PERCARD or UNEQUAL n SEQUENCED m PERCARD

This card indicates whether the cell frequencies are equal or unequal. n must be the number of observations per cell if cells are equal or the maximum number of observations per cell if cells are unequal.

The next three fields may be omitted if only one Data Card per cell is necessary. If, however, more than one Data Card per cell is necessary, the last three fields of this card must be included. m, in this case, is the number of data points per card.

No standard format is provided for this program; format must be supplied by the user as specified in the requirements for STPAC.

Data Cards

Data Cards are punched with code numbers that are the subgroup– or cell-identification numbers. These must be in integer format.

For data code = 2, the subgroup identification must be a single–integer number, and a single data point (in F, E, or D format) follows the identification number. Identification numbers in this data code need not be sequential (i.e., an analysis could be performed on subgroups identified as 1, 22, 1000, 50), but all cards for a single subgroup must be together.

For data codes = 0 and 1, Data Cards are punched with code numbers signifying level for each treatment, starting at the lefthand side of the card (cell identification). These must be in integer format. The first (I_1) indicates level for treatment 1; the second (I_2) indicates level for treatment 2; etc.

If the data code is 1, a single data point (in F, D, or E format) follows the cell identification (cell identification as in the analysis of variance). Data Cards with this data code may be in any order.

Data Code = 2 FORM: I X

where I is an integer subgroup-identification number and X is a single data point in F, E, or D format. This is the proper form whether cell frequencies are equal or unequal.

continued

Data Code = 1 FORM: $I_1, \cdots, I_N$ X

where $I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; X is a single data point.

Data Code = 0 FORM 1: $I_1, \cdots, I_N\ X_1, \cdots, X_n$

WHEN: cell frequencies are equal

all data for a cell on single card

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; $X_1, \cdots, X_n$ are the n data points as specified on the Equal Card.

FORM 2: $I_1, \cdots, I_N\ M\ X_1, \cdots, X_M$

WHEN: cell frequencies are *un*equal

data for a cell on single card

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (an integer) is the number of data points in that particular cell; $X_1, \cdots, X_M$ are the M data points.

FORM 3: $I_1, \cdots, I_N\ J\ X_1, \cdots, X_m$

WHEN: cell frequencies are equal

more than 1 card necessary for all data points in each cell

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; J (integer) is the card sequence number; $X_1, \cdots, X_m$ are the m data points per sequenced card, as specified on the Equal Card.

FORM 4: $I_1, \cdots, I_N\ M\ J\ X_1, \cdots, X_m$

WHEN: cell frequencies are *un*equal

more than 1 card necessary for all data points in each cell

$I_1, \cdots, I_N$ are integer cell-identification numbers, identifying one level for each of the N treatments; M (integer) is the number of data points in that particular cell and must be repeated for every card in that cell; J (integer) is the card sequence number; $X_1, \cdots, X_m$ are the m data points per sequenced card, as specified on the Unequal Card.

continued

Description of Output

For each subgroup (or cell), its variance, degrees of freedom, and identification are printed.

The chi-square value of the Bartlett's test, along with its correction factor, degrees of freedom, and associated probability are printed.

NOTE: A Bartlett's test cannot and will not be calculated if any of the subgroups have only one observation or a variance equal to 0.

SAMPLE INPUT

```
EXECUTE BARTL
CELL SUMMARY
TITLE POLAND CHINA PIGS
FORMAT(I1.9X.F3.0)
DATA CODE 1
TREATMENTS 1 LEVELS 8
BEGIN DATA
1     1     2.0
1     1     2.8
1     1     3.3
1     1     3.2
1     1     4.4
1     1     3.6
1     1     1.9
1     1     3.3
1     1     2.9
1     1     1.1
2     1     3.5
2     1     2.8
2     1     3.2
2     1     3.5
2     1     2.3
2     1     2.4
2     1     2.0
2     1     1.6
3     1     3.3
3     1     3.6
         .
         .
         .
8     1     2.4
8     1     3.0
8     1     1.5
END DATA
STOP
```

SAMPLE OUTPUT

```
                          BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCE

                                      POLAND CHINA PIGS

                    VARIANCE                 DEGREES OF FREEDOM                TREATMENT
                     0.9C933                          9                            1
                     C.49696                          7                            2
                     0.08889                          9                            3
                     C.10214                          7                            4
                     0.14667                          5                            5
                     C.C8000                          3                            6
                     0.39367                          5                            7
                     C.39000                          3                            8

            CORRECTION FACTOR IS        1.07399

BARTLETTS'S (EST OF HOMOGENEITY OF VARIANCE:
  CHI SQUARE =      17.8350   PRCBABILITY = 0.012737
             7 DEGREES OF FREECCM
  USE CAUTION IN INTERPRETATION IF ANY CELL FREQUENCIES ARE LESS THAN 3

CHI SQUARE IS SIGNIFICANT AT THE .C5 LEVEL
  DATA DO NOT SUPPORT THE HYPOTHESIS OF HOMOGENEITY OF VARIANCE
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time was 4 seconds on the central processor unit, at $0.11 per second.

Charge to user = computer time + network overhead
= $0.44 + network overhead

CONTENTS—(STPAC) BARTL

pages	
1	Identification & Abstract
3– 6	User Instructions
7– 8	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	CalComp Plotter Subroutines
CALLING NAME	See specific subroutine
INSTALLATION NAME	The University of Iowa University Computer Center
AUTHOR(S) AND AFFILIATION(S)	CalComp Modified by the University of Iowa
LANGUAGE	FORTRAN IV
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Available for use at the University of Iowa, but not for distribution
CONTACT	Mrs. Louise R. Levine, Program Librarian, Applications Programming, University Computer Center, The University of Iowa, Iowa City, Iowa 52240 Tel.: (319) 353-5580

FUNCTIONAL ABSTRACT

The CalComp plot routines are a series of subroutines used to obtain graphic output. They are designed for the following purposes:

1) providing values to scale data to fit the plotting area,
2) drawing identification symbols at plotted data points,
3) drawing connected lines between data points (when desired),
4) drawing and labeling axes at any desired orientation, and
5) selecting from a large number of characters, numbers, and special symbols, any of which can be drawn in any size and at any angular orientation anywhere on the plotting surface.

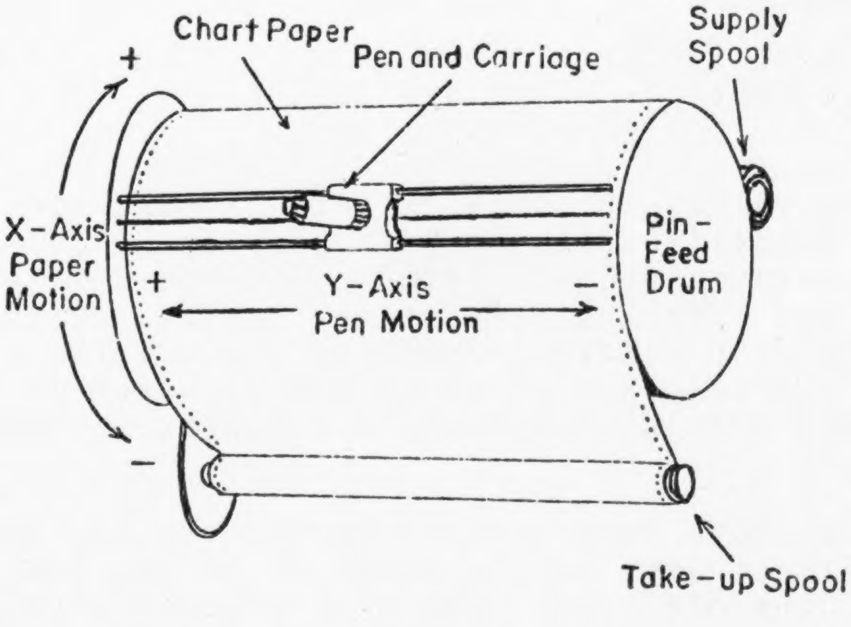
These routines have been written by CalComp and modified by the University of Iowa to run under OS on the IBM 360/65. Plotting is done off line. The user's plotter program will create a plot tape containing various commands used to drive a drum plotter (see fig. 1). Each of eight possible commands moves the pen 1/100 of an inch in one of the eight designated directions (see fig. 2). In addition there are commands to move

continued

the pen up and down. The plot tape is then taken to a smaller computer which drives the CalComp plotter (see fig. 3).

Access to the plot routines is by the FORTRAN "CALL" statement. Integer and floating point conventions must be followed (i.e. integer arguments must be used where the dummy argument specifies integer, likewise for floating point). All x and y coordinates used as arguments, must be expressed as floating point inches within actual page dimensions and in deflection from the (0.,0.) origin or the established reference point which is an established (0.,0.) origin. If the y coordinates are not within actual page dimensions in deflection from the reference point, that reference point will be destroyed and anything plotted after that will be erroneous. For orientation of the paper see fig. 1. All angles must be expressed in floating point degrees, with the positive sense counterclockwise. Character heights are specified as floating point inches. These heights should be greater than 0.07 but less than page size. Page size is 11 inches high and any reasonable length, since paper is continuous to a maximum length of 120 ft.

Figure 1



continued

Figure 2

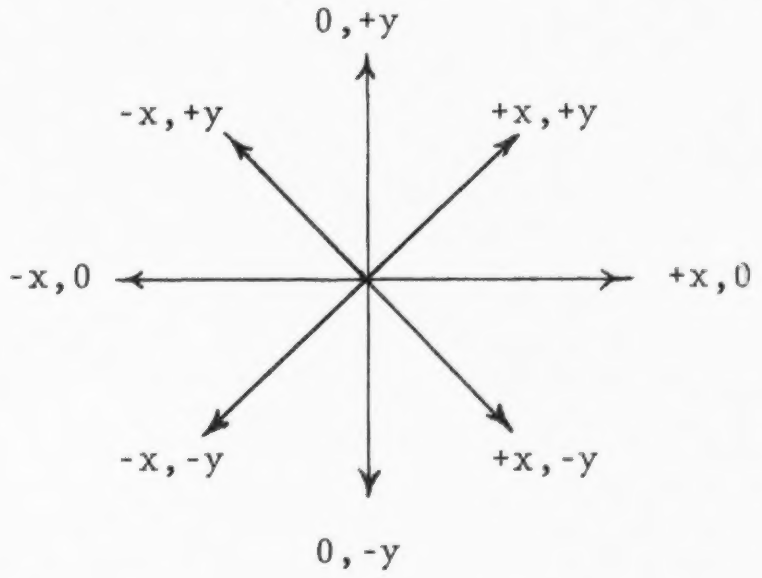
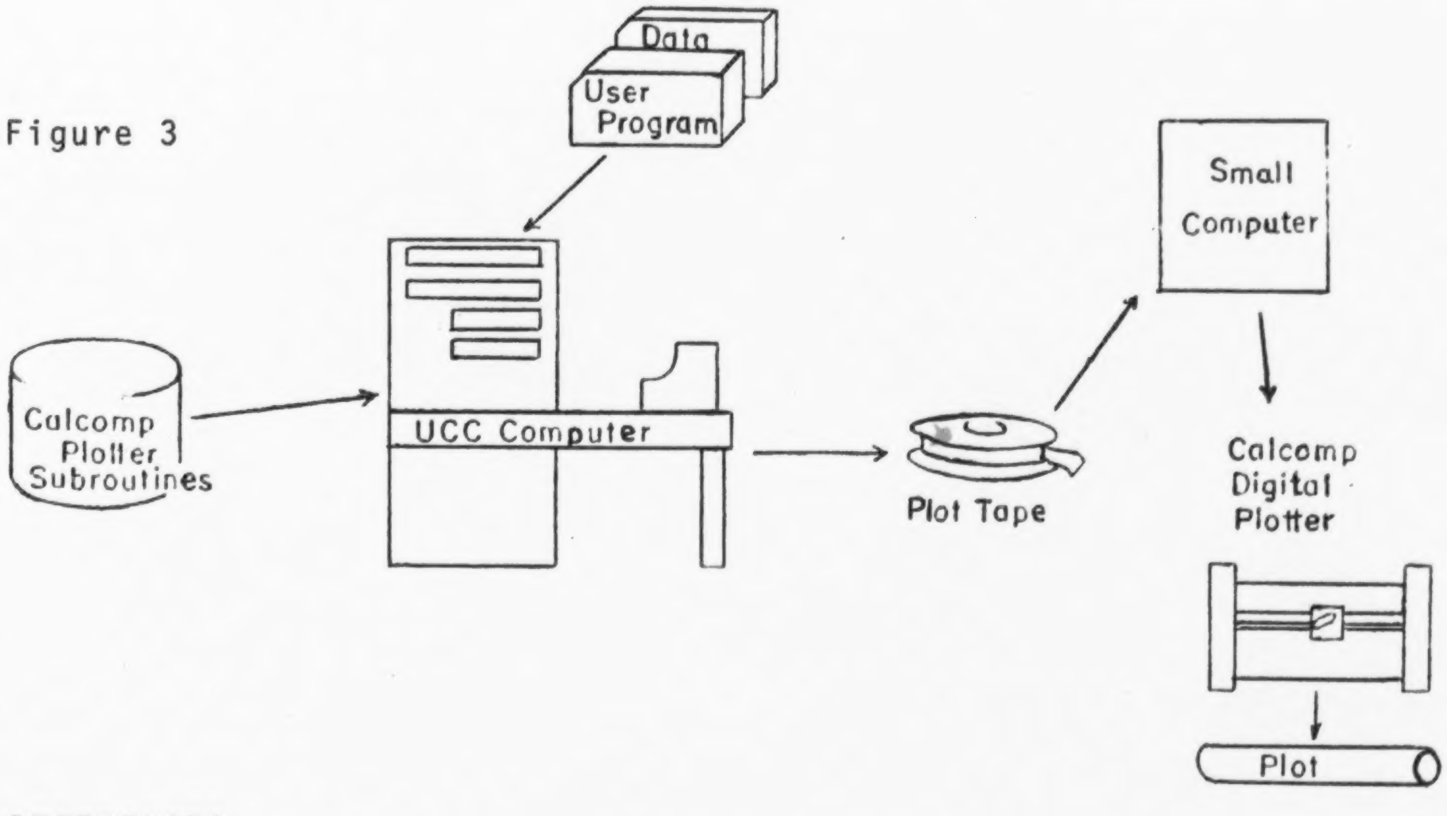

0,+y
-x,+y
+x,+y
-x,0
+x,0
-x,-y
+x,-y
0,-y

Figure 3


Data
User Program
Small Computer
Calcomp Plotter Subroutines
UCC Computer
Plot Tape
Calcomp Digital Plotter
Plot

REFERENCES

Plotter Subroutines for CalComp Digital Incremental Plotter (Iowa City, Ia.: Univ. Comp. Ctr., Univ. of Ia., 1970), Revised. Available from the EIN Office at the cost of reproduction and mailing.

USER INSTRUCTIONS

The user is referred to the reference below for instructions regarding use of the subroutines.

REFERENCES

Plotter Subroutines for CalComp Digital Incremental Plotter (Iowa City, Ia.: Univ. Comp. Ctr., Univ. of Ia., 1970), Revised. Available from the EIN Office at the cost of reproduction and mailing.

SAMPLE INPUT

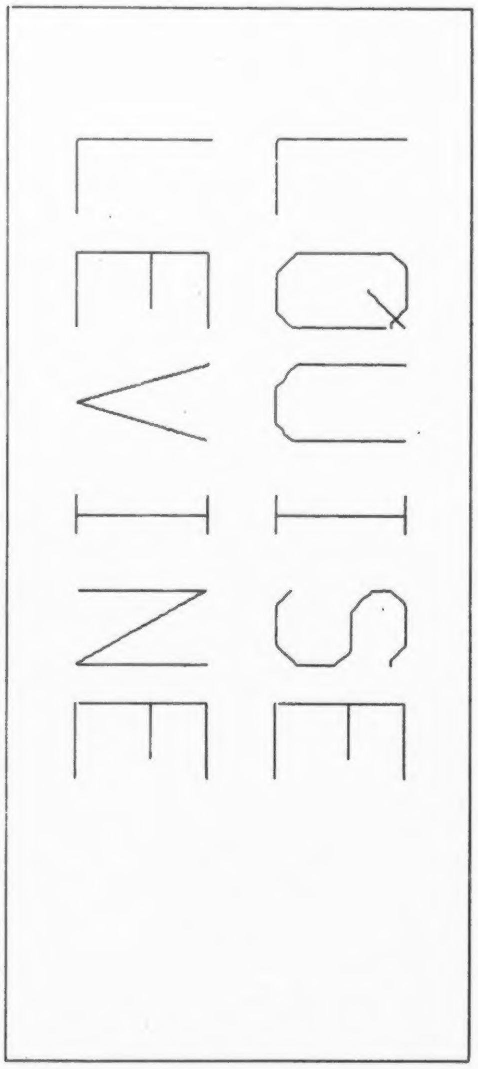
```
// EXEC FORTPLOT
//FORT.SYSIN DD *
      DIMENSION BUFFER(600),IDENT(9)
      DATA IDENT/'     LOUISE LEVINE  UCC PLOT COURSE '/
C SET REFERENCE POINT PENUP
      CALL PLOT(0.,-12.,-3)
C MOVE ORIGIN PEN UP
      CALL PLOT(0.,2.,-3)
C DRAW BOX 8X3.5 PEN DOWN
      CALL PLOT(8.,0.,2)
      CALL PLOT(8.,3.5,2)
      CALL PLOT(0.,3.5,2)
      CALL PLOT(0.,0.0,2)
C PRINT NAME IN BOX
      CALL SYMBOL(1.,2.,1.,'LOUISE',0.,6)
      CALL SYMBOL(1.,.5,1.,'LEVINE',0.,6)
C TERMINATE PLOTTING
      CALL WRAPUP
      CALL EXIT
      END
//GO.SYSIN DD *
/*
```

000 0052

SAMPLE OUTPUT

REDUCED TO 60% OF ORIGINAL SIZE.

LOUISE LEVINE UCC PLOT COURSE



COST ESTIMATE

The costing algorithm is based on the amount of computing time, I/O accesses, core storage, cards in and out, and lines and pages printed. The total cost for the Sample Output above was $0.71.

Charge to user = computer costs + postage + network overhead
= $0.71 + postage + network overhead

CONTENTS—CALCOMP PLOTTER SUBROUTINES

pages
1– 3 Identification & Abstract
5 User Instructions
7– 8 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	BEEF Data Processing Subroutines
CALLING NAME	BEEFDP
INSTALLATION NAME	The University of Notre Dame Computing Center
AUTHOR(S) AND AFFILIATION(S)	Westinghouse Electric Corporation UNIVAC Division of Sperry Rand Corporation
LANGUAGE	SLEUTH II/FORTRAN IV
COMPUTER	UNIVAC 1107
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Elizabeth Hutcheson, EIN Technical Representative, Computing Center, University of Notre Dame, Notre Dame, Ind. 46556 Tel.: (219) 283-7784

FUNCTIONAL ABSTRACT

The BEEF Data Processing library is a set of subroutines supplied by UNIVAC to "enhance FORTRAN's abilities as a commercial processor." This enrichment is in the form of subroutines for whole word data movement, character and field movement, supplementary formatting, decision-making with FORTRAN arrays, data conversion, report generation and control, I/O control, sorting, and compatability with EAM/EDP (Electronic Accounting Machine/Electronic Data Processing) equipment.

REFERENCE

UNIVAC 1107 BEEF Data Processing Manual, (UP-3985), (UNIVAC Division of Sperry Rand Corp., N.Y.). Copies of this manual are available through the local UNIVAC representative or through the EIN Office at the cost of reproduction and mailing.

USER INSTRUCTIONS

Full details of the description and use of the BEEFDP library are given in the reference cited below. A brief description of the library subroutines is presented as an aid for the interested person. The user should specify the subprogram name to obtain all the subroutines listed under that name, for a run.

Subprogram Name	*Subroutine Name*	*Description*
WWDM		**Whole Word Data Movement**
	MOVEKA	Moves the contents of a word into an array
	MOVEKL	Moves the contents of a word into a list specified in the calling sequence
	MOVEAA	Moves whole words from one array to another
	MOVEAL	Moves whole words from an array to a list specified in the calling sequence
	MOVELA	Moves whole words from a list specified in the calling sequence to an array
	MOVELL	Moves one list of words to another
CAFM		**Character and Field Movement**
	IXTRAC	Selects a character from an array and creates a variable FORTRAN integer
	INSERT	Replaces a character with the low order bits of a FORTRAN integer variable
	MOVECH	Transfers a field of characters from one array to another
	EXTRAC	Selects a character from an array and creates a Hollerith variable
	ENSERT	Places a single Hollerith character into an array

continued

Subprogram Name	*Subroutine Name*	*Description*
FORMAT		Formatting
	MOVMCH	Transfers several fields of characters from one array to another
	MOVECC	Packs or unpacks several fields of characters from one array to another
	MOVEHA	Defines Hollerith constants which may be used as FORTRAN variables
	HEDSET	Sets up to allow automatic writing of headings on reports
	HEDING	Allows the automatic writing of headings on reports
	CLKNDT	Transfers the date and/or time as FORTRAN integers
	DATBCD	Transfers the date as Hollerith characters
DECM		Decision Making with FORTRAN Arrays
	CMPAA	Compares logically or arithmetically a number of words of two arrays
	CMPAL	Compares logically or arithmetically a number of words of an array and a list
	CMPLL	Compares logically or arithmetically a number of words of two lists
	CMPKA	Determines if a word exists in a number of words of an array
	CMPKL	Determines if a word exists in a list
	FINDKA	Searches an array for a one word match
	FINDKL	Searches a list for a one word match
DATAC		Data Conversion
	BCD21	Converts a Hollerith number into a FORTRAN integer

continued

Subprogram Name	*Subroutine Name*	*Description*
	DECIN	Creates a FORTRAN integer from a Hollerith number (two word)
	DECN	Creates a FORTRAN integer from a Hollerith number (one word)
	AAADEC	Creates a signed Hollerith field from a FORTRAN integer
	AADEC	Creates a signed, edited Hollerith field from a FORTRAN integer
	LEDZER	Converts leading of a word to blanks
RPTGC		Report Generation and Control
	RPTFLT	Real variable report generator control
	RPTFIX	Integer variable report generator control
IOC		Input/Output Control
	BSFILE	Backspace a file
	UNLOAD	Rewind FORTRAN unit with interlock
	EOFBIN	End-of-file, binary
	EOFILE	End-of-file, Hollerith
	IOPACK	Blocking and buffering routine for fixed and variable tape records
	COPY	Copy a print file to the printer
SORTS		Sorting Routines
	BCDSR MLSRT	Internal commercial and arithmetic sort
COMPAT		Compatibility of EAM/EDP Equipment
	EAM	Creates a word containing the positional representation of the commercial sequence of the characters of a word

continued

Subprogram Name	*Subroutine Name*	*Description*
	EDP	Restores the word of the EAM routine
	CMPEAM	Compares two arrays for commercial sequence
	KMPWL	Compares word fields in two arrays for Field Data sequence
	KMPWA	Compares word fields in two arrays for Algebraic sequence
	KMPCL	Compares character fields in two arrays for Field Data sequence
	KMPCA	Compares character fields in two arrays for Algebraic sequence
	KMPWC	Compares word fields in two arrays for commercial sequence
	KMPWS	Compares word fields in two arrays for scientific sequence
	KMPCC	Compares character fields in two arrays for commercial sequence
	KMPCS	Compares character fields in two arrays for scientific sequence
	LOGSET	Compares (EAM) two strings of Hollerith characters
	JC	Defines a FORTRAN constant
	JD	Divides one FORTRAN constant by another
	JM	Multiplies one FORTRAN constant by another
	ENDFIL	Mark end-of-file on FORTRAN unit
	REWIND	Rewinds FORTRAN unit

REFERENCE

UNIVAC 1107 BEEF Data Processing Manual, (UP-3985), (UNIVAC Division of Sperry Rand Corp., N.Y.). Copies of this manual are available through the local UNIVAC representative or through the EIN Office at the cost of reproduction and mailing.

SAMPLE INPUT and SAMPLE OUTPUT

Sample Input and Output are included in the reference listed under the USER INSTRUCTIONS. Interested persons are directed to this source.

COST ESTIMATE

See reference listed in USER INSTRUCTIONS.

Charge to user = computer time + postage and handling + network overhead

= computer time + $10.00 + network overhead

CONTENTS—BEFFDP

pages

1 Identification & Abstract
3– 6 User Instructions
7 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	BEEF Mathematical Subroutine
CALLING NAME	BEEFM
INSTALLATION NAME	University of Notre Dame Computation Center
AUTHOR(S) AND AFFILIATION(S)	UNIVAC Division of Sperry Rand Corporation Westinghouse Electric Corporation Baltimore Defense and Space Center Boeing Corporation
LANGUAGE	FORTRAN IV and SLEUTH II
COMPUTER	UNIVAC 1107
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mrs. Elizabeth Hutcheson, Computer Center, University of Notre Dame, Notre Dame, Ind. 46556 Tel.: (219) 283-7784

FUNCTIONAL ABSTRACT

The BEEF mathematical library is a set of subroutines supplied by UNIVAC to "enhance FORTRAN's abilities as a scientific processor." This enrichment is in the form of subroutines for the evaluation of mathematical functions, matrix arithmetic, and other standard engineering requirements.

See the User Instructions for further information on BEEFM.

REFERENCE

UNIVAC 1107 BEEF Math Routines Manual (UP-3984), UNIVAC Division of Sperry Rand Corporation, (New York, N.Y., March 1965). Available through the UNIVAC 1107 Librarian, Systems Prog. Library Services, UNIVAC Div. of Sperry Rand, Sperry Rand Bldg., New York, N.Y. 10019.

USER INSTRUCTIONS

Program Description

The following subroutines comprise the BEEF Math library.

Name	*Description*
DED	SOLUTION OF A SET OF FIRST ORDER DIFFERENTIAL EQUATIONS The subroutines DED, DEI, DES and DEDIS solve a system of N first-order ordinary differential equations having the form $Y_i^1 = f_i(t, y_1, y_2, \cdots y_n)$ The user may select one of the following options. (a) a fourth order Runge-Kutta method with fixed step size (b) Adams-Moulton method with fixed step size (c) Adams-Moulton method with variable step size
LAGIT	LAGRANGIAN INTERPOLATION This subroutine computes y = f(x) given a discrete set of values of x and y, using variable degree Lagrangian interpolation by the following formula. $P_n(x) = \sum_{s=1}^{n} L_s(x) f(x_s)$ where $L_s(x) = \prod_{\substack{t=1 \\ t \neq s}}^{n} \frac{(x-x_t)}{(x_s-x_t)} \quad \text{for } s = 1, 2, \cdots, n$ and $Y_s = f(x_s)$ are known for x_s.
RANDM	UNIFORM OR NORMAL NUMBER GENERATOR This subroutine computes either a uniform or normally distributed random number.
RKVS	RUNGE-KUTTA VARIABLE STEP This subroutine solves numerically a set of N first-order differential equations (linear or non-linear) using a fifth-order Runge-Kutta

continued

Name	*Description*
	technique. Speed and accuracy are attained by using a step size control. However, because of the necessity to make at least twelve evaluations of each functional expression, complicated functions or large sets of functions may consume a large amount of time particularly if the values of the dependent variables are desired at a number of points and/or the interval of integration is large.
SPLIN	SPLINE INTERPOLATION This set of subroutines fits a set of points with an interpolating function made up of piecewise curves which act somewhat as an ideal spline. First and second derivatives of the fitted curve are continuous from interval to interval. The user must specify either the first or the second derivatives (but not both) at the end points. The condition that the second derivative be zero at an end means that the end is not held in a bent position.
SETU	GAUSS QUADRATURE This subroutine provides the abscissa and weight coefficients used in the evaluation of a definite integral by Gauss's mechanical quadrature formula. This formula differs from Simpson's Rule and Weddle's Rule in that the subdivisions are not equidistant but are symmetrically placed with respect to the midpoint of the interval of integration.
COMBES	COMPLEX BESSEL FUNCTION (COMPLEX ARGUMENT AND ORDER) This subroutine computes the Bessel Functions, $J_{a\pm n+ib}(x+iy)$ and $Y_{a\pm n+ib}(x+iy)$ of complex order and argument. The method used is that of Goldstein and Thaler.[2]
NSIMEQ	SIMULTANEOUS LINEAR EQUATION The subroutine NSIMEQ solves the matrix equation $A \cdot X=B$ for the unknown matrix X. The subroutine NDETRM computes a scaled value of the determinant of the A matrix.
ORTHLS	ORTHOGONAL POLYNOMIAL LEAST SQUARES CURVE FITTING This subroutine finds the polynomial $P_k(x)$ of degree

continued

Name	*Description*
	$\leq K$ which best approximates, in the least squares sense, a weighted set of data points using orthogonal polynomials. The fitted values for a given set of arguments and the fitted value and derivative for a single argument may also be obtained. The user must provide input and output in source program.
GAMMA	GAMMA FUNCTION This subroutine calculates the value of $\Gamma(1 + X)$.
TABF	SINGLE INTERPOLATION This subroutine performs a K-th order interpolation ($K<6$) on a table of X_iY_i values giving, for a given value of X, the corresponding value of Y. The method of interpolation used by this subroutine is Aitken's method as described in Ref. 3. Independent variables must be monotonic but need not be equally spaced.
BES	BESSEL FUNCTION (REAL ARGUMENT AND ORDER) This subroutine computes the Bessel functions for real argument and order. The method used is that of Goldstein and Thaler.[2]
MATAB	MATRIX ABSTRACTION This subroutine performs addition, subtraction, multiplication, postmultiplication and transposition of real and complex matrices.
MATRTA	MATRIX ROTATION This subroutine will rotate a 3 X 3 matrix about any one of the three orthogonal axes through any given angle.
CINV1	MATRIX INVERSION AND DETERMINANT EVALUATION OF COMPLEX MATRICES This subroutine provides for inversion and determinant evaluation of general complex matrices or real matrices compatible with existing real matrix subroutine. The matrix is inverted by a variation of the Gauss elimination method.
INV	MATRIX INVERSION AND DETERMINANT EVALUATION (REAL) This subroutine inverts an N X N real matrix or

continued

Names	*Description*
	evaluates its determinant. The mathematical technique employed for matrix is a variation of the Gauss elimination method.
DETIGN	DETERMINANT AND NORMALIZED EIGENVECTORS OF A COMPLEX MATRIX This subroutine calculates the determinant and (if desired) normalized eigenvectors of a complex matrix.
TBLP	SINGLE LINEAR INTERPOLATION LOOK-UP This subroutine performs a single linear interpolation. The interpolation tables must have previously read into the computer. The set of values for the independent variable must be monotonically increasing.
GDT	SIMULTANEOUS LINEAR EQUATIONS (SINGLE OR DOUBLE PRECISION) This subroutine solves a set of N single or double precision simultaneous linear equations.
INV1	MATRIX INVERSION AND DETERMINANT EVALUATION (SINGLE OR DOUBLE PRECISION) This subroutine inverts an N X N real matrix and evaluates its determinant.
CMPXMD	MULTIPLICATION AND DIVISION OF COMPLEX NUMBER OF LARGE MAGNITUDE The subroutines COMMPY and COMDIV perform a complex multiplication and division respectively of large magnitude numbers.
UDRNRT	UNIFORMLY DISTRIBUTED PSEUDO RANDOM NUMBER GENERATOR This subroutine produces a set of uniformly distributed numbers on the unit interval.
TBLF	PARABOLIC INTERPOLATION Given paired values of X and Y, this subprogram calculates the Y value for any particular value of X,XBAR within the range of the given X values.
PMTR	POLYNOMIAL MATRIX TRIANGULARIZATION ROUTINE This subroutine reduces a given polynomial matrix to triangular form. The subroutine saves computer time over the usual method of determinants by treating polynomial elements individually.

continued

Name	*Description*
ICEM	INTEGRATION WITH CONTROLLED ERROR This subroutine provides the numerical solution of an N–th order system of linear or non-linear differential equations expressed as a system of N first-order equations.
ARTN	FOUR QUADRANT ARCTANGENT SUBROUTINE This subroutine computes the arctangent of the ratio of two floating point numbers.
DEFI	DEFINITE INTEGRAL EVALUATION This subroutine computes the definite integral of functions which are not well behaved. Two estimates of the value of the integral are made, a parabolic approximation, and one using Simpson's Rule. The program will vary the step size to keep the error estimated within specified bounds. In specifying the minimum step size allowed, the user can greatly determine the capabilities of the program in handling oscillating or discontinuous functions.
ERF3	ERROR AND NORMAL FREQUENCY FUNCTION ERF3 consists of two subroutines, the error function and the normal frequency function.[4] For a given value, Y, the functions are evaluated using approximations for the following integrals $\mathrm{erf}(Y) = \frac{2}{\sqrt{\pi}} \int_0^Y e^{-t^2}\, dt \cong 1-(\sum_{i=0}^{6} A_i Y^i)^{-16}$ $F(Y) = \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{Y} e^{-t^2/2}\, dt \cong \frac{1}{2}[1 + \mathrm{ERF}(Y/\sqrt{2})].$
WEGSTN	WEGSTEIN ITERATION SUBROUTINE Given an implicit equation of the form X = f(X), the subroutine finds the value for X which provides a specified accuracy in either a relative or absolute sense. The method used is a modification of iteration which determines the root of the explicit function g(X)=0 by the iteration process $X_{n+1} = X_n - g(X)/g'(X).$ $X_{n+1} = \frac{f(X_n) - X_{nF'}}{1 - F'}$

continued

Name	*Description*
BESMMM	BESSEL FUNCTION EVALUATION This subroutine will evaluate the Bessel functions for a given floating point argument. Calculation is terminated either when the n^{th} computed value is equal to the n–1 value or when the n^{th} computed value is less than the absolute value of a preassigned test number.
RANNUM	NORMALLY DISTRIBUTED PSEUDO-RANDOM NUMBER GENERATOR This program will produce a normally-distributed random number with a given mean and standard deviation.
XLAGIN	LAGRANGE INTERPOLATION SUBROUTINE Given a table of paired values of X, Y, this program computes a value of Y corresponding to a specific XX value within the range of the table. The Lagrange interpolation formula is used. $Y = \sum_{k=1}^{n+1} \left[Y_k \prod_{\substack{j \\ j \neq k}} \frac{XX-X_j}{X_k-X_j} \right]$ The user must provide input and output routines in a source program. Data for the table need not be in descending order nor equally spaced.
SIM2	DEFINITE INTEGRAL EVALUATION This subroutine will compute the definite integral of a function using the maximum step size which will provide accuracy within a given bound. The program makes use of Simpson's Rule, computing and comparing two successive terms. Discontinuous functions or functions that oscillate rapidly may result in incorrect results.
DITTO	SPECIALIZED DOUBLE INTEGRAL EVALUATION This subroutine will evaluate the special definite integral $\int_{yL}^{yu} g(Y)\ dy \int_{xL=y}^{xu} f(X,Y)dx$ where g(Y) and f(X,Y) are defined by function subprograms; yu, yL and xu are fixed limits, and xL is a variable limit equal to y. The user may specify

continued

Name	*Description*
	the accuracy desired. The basic method of integration used is a variable-interval Simpson's Rule, modified to yield results two orders higher in accuracy than the basic Simpson's Rule.
FRENEL	FRESNEL INTEGRATION ROUTINE This subroutine evaluates the integral $(2\pi)^{-1} \int_0^x \frac{e^{-it}}{\sqrt{t}} dt$ for any value X>0. The output of the subroutine consists of the real and imaginary results of the integration.
FSERIE	FRACTION SERIES SOLUTION COMPLEX This subroutine calculates the N complex eigenvalues of a complex or real matrix, given N+1 distinct guesses, $\lambda_1, \lambda_2, \cdots, \lambda_n$, of the roots and the corresponding N+1 determinants, $D(\lambda_1), D(\lambda_2), \cdots, D(\lambda_{n+1})$. This subroutine may also be used to solve for the roots of the polynomial F(x). In this case, the guesses X_i of the roots replace the λ_i and $F(X_i)$ replaces $D(\lambda_i)$ for $i=1,2,\cdots,n+1$.
INTEG	SINGLE OR DOUBLE INTEGRAL EVALUATION This subroutine will evaluate the double definite integral $\int dy \int f(X,Y)dx$ where f(X,Y) and the limits of integration are defined by the user. The accuracy is also user-specified. The basic method of integration used is a variable-interval Simpson's Rule, modified to yield results two orders higher in accuracy than the basic Simpson's Rule. This program was designed primarily for functions which can be expressed mathematically, rather than tabulated data.
PLOTXX	PLOT ORDERING SUBROUTINE This subroutine will provide a graphical representation of data stored in computer memory for listing on a standard printer. A maximum of 20 curves may be depicted. Scaling is automatic and all titles and plotting symbols may be specified by the user.

continued

Name	*Description*
ROOTS	ROOTS OF A POLYNOMIAL This subroutine obtains approximations to the N complex roots of a polynomial with complex coefficients by the Newton-Raphson method. The polynomial must be of degree ≤ 50.

Description of Input

For a complete description of the options available and the usage conventions the user should consult the reference cited below.

Control Cards

Run Card

The Run Card will be provided by University of Notre Dame personnel.

ASG Card

Columns	*Contents*
1	Multipunch 7 and 8
2–12	ASG A=BEEFM

PLT Card

Columns	*Contents*
1	Multipunch 7 and 8
2– 6	PLT, A
8–72	The user should list the names of the subprograms he desires, separated by commas.

Note: The subprogram name appearing in the left column of the description given in the Functional Abstract should be specified. In this way, all the subroutines comprising a subprogram will be available.

FOR Card

Columns	*Contents*
1	Multipunch 7 and 8
2– 4	FOR

continued

Columns	*Contents*
5	Blank
6–10	User-supplied job name

XQT Card

Columns	*Contents*
1	Multipunch 7 and 8
2– 4	XQT
5	Blank
6–10	Job name specified on FOR Card

FIN Card

Columns	*Contents*
1	Multipunch 7 and 8
2– 4	FIN

Input Deck

Run Card
ASG Card
PLT Card
FOR Card
    [Program Deck]
XQT Card
    [Data Cards, if any]
FIN Card

REFERENCES

1. UNIVAC 1107 BEEF Math Routines Manual (UP–3984), UNIVAC Division of Sperry Rand Corporation, (New York, N.Y., March 1965). Available through the UNIVAC 1107 Librarian, Systems Prog. Library Services, UNIVAC Div. of Sperry Rand, Sperry Rand Bldg., New York, N.Y. 10019.

2. Goldstein, M., and Thaler, M.R., "Recurrence Techniques for the Calculation of Bessel Functions," MATC *XIII*, 1959, p. 102.

3. Milne, W.E., *Numerical Calculus*, (Princeton University Press, Princeton, N.J., 1949).

4. Hildebrand, F.B., *Introduction to Numerical Analysis*, (McGraw-Hill Book, Co., Inc., New York, N.Y., 1956) p. 3.

SAMPLE INPUT (FOR GAMMA)

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEEFM
- PLT,A GAMMA
- FOR TEST
C     MAIN PROGRAM TO TEST GAMMA FUNCTION ROUTINE
C
      DIMENSION X(50)
C
C     PROGRAM INITIALIZATION.
C
      WRITE (6,100)
    1 READ (5,101) N
      READ (5,102) (X(I),I=1,N)
C
C     CALCULATE GAMMA FUNCTION.
C
      DO 2 I=1,N
      Y=GAMMA(X(I))
C
C     OUTPUT.
C
    2 WRITE (6,103) X(I),Y
      GO TO 1
C
C     FORMAT STATEMENTS.
C
  100 FORMAT (45H1MAIN PROGRAM TO TEST GAMMA FUNCTION ROUTINE.////)
  101 FORMAT (I10)
  102 FORMAT (F10.0)
  103 FORMAT (8H FOR X = F15.7,10X,17HGAMMA FUNCTION = F15.7//)
      END
- XQT TEST
         9
       2.3
       2.0
       1.3
       1.0
       0.0
       0.7
       1.7
       2.7
       2.0
- FIN
```

000 0054

SAMPLE OUTPUT (FROM GAMMA)

```
MAIN PROGRAM TO TEST GAMMA FUNCTION ROUTINE.


FOR Y =      2.3000000        GAMMA FUNCTION =       2.6834374

FOR Y =      2.0000000        GAMMA FUNCTION =       2.0000000

FOR Y =      1.3000000        GAMMA FUNCTION =       1.1667120

FOR Y =      1.0000000        GAMMA FUNCTION =       1.0000000

FOR Y =       .0000000        GAMMA FUNCTION =       1.0000000

FOR Y =     -.7000000         GAMMA FUNCTION =       2.9015490

FOR Y =    -1.7000000         GAMMA FUNCTION =      -4.2736701

FOR Y =    -2.7000000         GAMMA FUNCTION =       2.5139236



ATTEMPTED TO TAKE GAMMA FUNCTION OF   -.2000000+01
```

SAMPLE INPUT (FOR BES)

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEFFM
- PLT,A BES
- FOR TEST
C      MAIN PROGRAM TO TEST BESSEL FUNCTION SUBROUTINE FOR REAL ARGUMENT
C      AND ORDER.
C
       DIMENSION BJ(250),BY(250),BI(250),BK(250),TX(20),TFNU(20),IN(10)
C
C      PROGRAM INITIALIZATION.
C
       WRITE (6,100)
     1 READ (5,110) NX,NU,NON
       READ (5,111) (TX(I), I=1,NX)
       READ (5,111) (TFNU(I), I=1,NU)
       READ (5,110) (IN(I), I=1,NON)
C
C      LOOP TO TAKE ALL POSSIBLE VALUES, ONE FROM EACH TABLE.
C
       DO 10 I=1,NX
       DO 10 J=1,NU
       DO 10 K=1,NON
       X=TX(I)
       FNU=TFNU(J)
       N=IN(K)
       N1=IABS(N)+1
C
C      CALLING SEQUENCE.
C
       CALL BESY(X,FNU,N,BJ,BY)
       GO TO 2
     3 CALL BESK (X,FNU,N,BI,BK)
       GO TO 4
C
C      OUTPUT.
C
     2 WRITE (6,102) X,FNU,N
       WRITE (6,103) (BJ(L),L=1,N1)
       WRITE (6,104) (BY(L),L=1,N1)
       GO TO 3
     4 WRITE (6,105) (BI(L), L=1,N1)
       WRITE (6,106) (BK(L), L=1,N1)
    10 CONTINUE
       GO TO 1
C
C      FORMAT STATEMENTS.
C
   100 FORMAT (77H1MAIN PROGRAM TO TEST BESSEL FUNCTION SUBROUTINE FOR RE
      1AL ARGUMENT AND ORDER.////)
   102 FORMAT (5X2HX=E14.7,5X,3HNU=E14.7,5X,2HN=I5//)
   103 FORMAT(17H0OUTPUT FROM BESJ//(5E17.8))
   104 FORMAT(17H0OUTPUT FROM BESY//(5E17.8))
   105 FORMAT(17H0OUTPUT FROM BESI//(5E17.8))
   106 FORMAT(17H0OUTPUT FROM BESK//(5E17.8))
   110 FORMAT (14I5)
   111 FORMAT (4F15.0)
       END
· XQT TEST
    8    1    2
              3.             4.             6.             7.
              8.       2.404826       3.957678       7.015587
              0.
    9   -9
FIN
```

SAMPLE OUTPUT (FROM BES)

```
MAIN PROGRAM TO TEST BESSEL FUNCTION SUBROUTINE FOR REAL ARGUMENT AND ORDER.


    X=   .3000000+01      NU=   .0000000        N=    9


OUTPUT FROM BESJ

   -.26005197-00     .33905807-00     .48609128-00     .30906273-00     .13203419-00
    .43028437-01     .11393937-01     .25472946-02     .49344181-03     .84395030-04

OUTPUT FROM BESY

    .37685003-00     .32467478-00    -.16040044-00    -.53854163-00    -.91668281-00
   -.19059850+01    -.54764701+01    -.10839934+02    -.87149888+02    -.44408947+03

OUTPUT FROM BESI

    .24300037-00     .19682673-00     .11178256+00     .47783325-01     .16215908-01
    .45409037-02     .10795605-02     .22265772-03     .40511860-04     .65904643-05

OUTPUT FROM BESK

    .69776157-00     .80656378-00     .12354705+01     .24538573+01     .61431850+01
    .18875687+02     .68028707+02     .20455087+03     .14434094+04     .79932146+04
    X=   .3000000+01      NU=   .0000000        N=   -9


OUTPUT FROM BESJ

   -.26005197-00    -.33905807-00     .48609128-00    -.30906274-00     .13203420-00
   -.43028448-01     .11393941-01    -.25473940-02     .49387803-03    -.86622138-04

OUTPUT FROM BESY

    .37685003-00    -.32467478-00    -.16040044-00     .53854163-00    -.91668281-00
    .19059850+01    -.54764701+01     .10839934+02    -.87149888+02     .44408947+03

OUTPUT FROM BESI

    .24300037-00     .19682677-00     .11178255+00     .47783326-01     .16215903-01
    .45409197-02     .10795053-02     .22289814-03     .39313078-04     .13223598-04

OUTPUT FROM BESK

    .69776157-00     .80656378-00     .12354705+01     .24538573+01     .61431850+01
    .18875687+02     .68028707+02     .20455087+03     .14434094+04     .79932146+04
    X=   .4000000+01      NU=   .0000000        N=    9


OUTPUT FROM BESJ

   -.39714982-00    -.66043326-01     .36412816-00     .43017148-00     .28112907-00
    .13208666-00     .49087576-01     .15176070-01     .40286678-02     .93860185-03

OUTPUT FROM BESY

   -.16940730-01     .39792570-00     .21590358-00    -.18202212-00    -.48803676-00
   -.79559140-00    -.15006917+01    -.37062238+01    -.11471092+02    -.42178143+02

OUTPUT FROM BESI

    .20700198-00     .17875096-00     .11762652+00     .61124345-01     .25930997-01
    .92443600-02     .28391218-02     .75698479-03     .17967511-03     .38284325-04
```

continued

```
OUTPUT FROM BESK

     .60929767-00     .68157584-00     .95008555-00     .16316414+01     .33975776+01
     .84268164+01     .24464618+02     .81820672+02     .31083496+03     .13251685+04
     X=   .4000000+01       NU=   .0000000           N=   -9


OUTPUT FROM BESJ

    -.39714982-00     .66043396-01     .36412816-00    -.43017148-00     .28112907-00
    -.13208665-00     .49087575-01    -.15176062-01     .40286426-02    -.93860866-03

OUTPUT FROM BESY

    -.16940730-01    -.39792570-00     .21590358-00     .18202212-00    -.48893676-00
     .79585140-00    -.15006917+01     .37062238+01    -.11471092+02     .42178143+02

OUTPUT FROM BESI

     .20700198-00     .17875086-00     .11762651+00     .61124345-01     .25940996-01
     .92443524-02     .28291105-02     .75700706-03     .17958739-03     .38658269-04

OUTPUT FROM BESK

     .60929767-00     .68157584-00     .95008555-00     .16316414+01     .33975776+01
     .84268164+01     .24464618+02     .81820672+02     .31083496+03     .13251685+04
     X=   .6000000+01       NU=   .0000000           N=    9


OUTPUT FROM BESJ

     .15064526-00    -.27668385-00    -.24287323-00     .11476837+00     .35764159-00
     .36208705-00     .24583687-00     .12958646-00     .56531096-01     .21165325-01

OUTPUT FROM BESY

    -.28819468-00    -.17501031-00     .22985702-00     .32824893-00     .98390996-01
    -.19706097-00    -.42682507-00    -.65659082-00    -.11052194+01    -.22906607+01

OUTPUT FROM BESI

     .16665758-00     .15205155-00     .11597369+00     .74735767-01     .41238923-01
     .19751870-01     .83181300-02     .31155905-02     .10484288-02     .31978041-03

OUTPUT FROM BESK

     .50186317-00     .54217576-00     .68258821-00     .99723406-00     .16798223+01
     .32369971+01     .70748177+01     .17386631+02     .47643623+02     .14443628+03
     X=   .6000000+01       NU=   .0000000           N=   -9


OUTPUT FROM BESJ

     .15064526-00     .27668385-00    -.24287323-00    -.11476837+00     .35764160-00
    -.36208705-00     .24583688-00    -.12958667-00     .56532010-01    -.21165356-01

OUTPUT FROM BESY

    -.28819468-00     .17501031-00     .22985702-00    -.32824893-00     .98390994-01
     .19706097-00    -.42682507-00      65659082-00    -.11052194+01     .22906607+01

OUTPUT FROM BESI

     .16665758-00     .15205155-00     .11597369+00     .74735768-01     .41238921-01
     .19751874-01     .83181302-02     .31156095-02     .10483768-02     .31907812-03

OUTPUT FROM BESK

     .50186317-00     .54217576-00     .68258821-00     .99723406-00     .16798223+01
     .32369971+01     .70748177+01     .17386631+02     .47643623+02     .14443628+03
     X=   .7000000+01       NU=   .0000000           N=    9
```

continued

```
OUTPUT FROM BESJ

     .30007628-00    -.46028008-00    -.30141722-00    -.16755560-00     .15770813-00
     .34789437-00     .33019641-00     .23358758-00     .12797054-00     .58900513-01

OUTPUT FROM BESY

    -.25949737-01    -.30066703-00    -.60526614-01     .26808060-00     .29030998-00
     .63702226-01    -.15030679-00    -.40537102-00    -.61143524-00    -.99210523-00

OUTPUT FROM BESI

     .15373613-00     .14028959-00     .11308396+00     .77670165-01     .46500517-01
     .24516456-01     .11486015-01     .40261015-02     .18337322-02     .63475376-03

OUTPUT FROM BESK

     .46584500-00     .45506905-00     .60815059-00     .84558387-00     .13390367+01
     .23689401+01     .47171348+01     .10455460+02     .25628057+02     .69033875+02
     Y=   .800000+01      NU=   .0000000          N=   -9


OUTPUT FROM BESJ

     .30007628-00     .46028008-00    -.30141722-00     .16755560-00     .15770813-00
    -.34789437-00     .33019641-00    -.23358757-00     .12797052-00    -.58900491-01

OUTPUT FROM BESY

    -.25949737-01     .30066703-00    -.60526614-01    -.26808060-00     .29030998-00
    -.63702926-01    -.15030679-00     .40537102-00    -.61143524-00     .99210523-00

OUTPUT FROM BESI

     .15373813-00     .14028959-00     .11308396+00     .77670185-01     .46500519-01
     .24516446-01     .11486001-01     .40261280-02     .18337649-02     .63466530-03

OUTPUT FROM BESK

     .46584500-00     .45506905-00     .60815059-00     .84558387-00     .13390367+01
     .23689401+01     .47171348+01     .10455460+02     .25628057+02     .69033875+02
     Y=   .800000+01      NU=   .0000000          N=    9


OUTPUT FROM BESJ

     .17165070-00     .23463632-00    -.11299171+00    -.29113218-00    -.10535742+00
     .18577474-00     .33757507-00     .32058905-00     .22345496-00     .12632088-00

OUTPUT FROM BESY

     .22352140-00    -.15806051-00    -.26303662-00     .26542198-01     .28204326-00
     .25640104-00     .37558048-01    -.20006397-00    -.38767000-00    -.57527605-00

OUTPUT FROM BESI

     .14343173-00     .13414244-00     .10989612+00     .79194386-01     .50500330-01
     .28694056-01     .14432741-01     .67449155-02     .28291585-02     .10865986-02

OUTPUT FROM BESK

     .43662307-00     .46714941-00     .55241038-00     .73935460-00     .11060263+01
     .18462806+01     .34147734+01     .60684470+01     .15609560+02     .38187566+02
     Y=   .800000+01      NU=   .0000000          N=   -9
```

continued

```
OUTPUT FROM FESI

     .143"3,7*-00      .13"14<""-0^      •10989612+00      •*9194*86-01      .505^^330-01
     •286^"0^5"-01     .1^*327*^-^1      •67449121-02      •^8291^58-02      .108^^806-02

OUTPUT FROM FESK

     •436^2*0*-00      .4^*145"1-0^      •55241^*^-00      •*3935"60-00      .110^^263+01
     .1846?^0^+01      .3^1477*"+01      •60684^70+01      •15609^60+02      .381^7566+02
     Y=    .^4^4626+01       NU=    .000^0^0          N=     0


OUTPUT FROM FESJ

   -.239717^1-06       .51^147"1-0^      •4*175"^^-00      •19899099-00      .647^^709-01
     .163^9^5*-01      .3^^48^^1-0^      •60068^1"-03      •^2165^22-04      .1251*292-04

OUTPUT FROM FESY

     .509^2^3^-00      .10*74^^^+0^     -•4^447378-00     -•^0878*46-00     -.159*"272+01
   -.44919^1^+01      -.17^65^^1+0^     -•807645^^+02     -•"5309^70+03     -.293*^116+04

OUTPUT FROM FESI

     .276^^^0^-00      .20^390^6-0^      •102983^7+00      •*7095^57-01      .104*0638-01
     .239^6*7^-02      .46"689^^-0*      •7*8503^1-04      •11474^23-04      .150^^421-05

OUTPUT FROM FESK

     .773*005"-00      .921776*8-0^      •15399077+01      •*4831"07+01      .102*^284+02
     .375154"5"+02     .1^^231^1+0*      •8^700^7*+0*      •^2136^16+04      .355^"732+05
     Y=    .^4^4626+01       NU=    .000^0^0          N=    -0


OUTPUT FROM FESJ

   -.239*1^01-06      -.51^147"1-0^      •4*175"^^-00     -•19899099-00      .647^^704-01
   -.163^9^"^-01       .34^475^*-0^     -•^00358^0-03      •^0312^38-04     -.513^^970-06

OUTPUT FROM FESY

     .509^2^3^-00     -.10*74^^^+0^     -•4^447378-00      •^0878*46-00     -.159*"272+01
     .44919^1^+01     -.17^856^1+0^      •807645^^+02     -•"5309^70+03      .293*^116+04

OUTPUT FROM FESI

     .276^^^0^-00      .20^390^6-0^      •102983^7+00      •*7095^55-01      .104*0645-01
     .239^611^-02      .46"603^5-0*      •7*25707^-04      •15041765-04     -.228^^119-04

OUTPUT FROM FESK

     .773*005"-00      .921776*8-0^      •15399077+01      •*4831"07+01      .102*^284+02
     .375154"5"+02     .1^^231^1+0*      •8^700^7*+0*      •^2136^16+04      .355^"732+05
     Y=    .*9^7678+01       NU=    .000^0^0          N=     9


OUTPUT FROM FESJ

   -.399^0^0^-00      -.49*970^^-0^      •3*443729-00      •42823^41-00      .2747^947-00
     .127^1-5^-00      .4^^553^^-0^      •1"2453**-01      •*7364*90-02      .860"^559-03

OUTPUT FROM FESY

     .158*2558-06      .40^542*0-0^      •20342351-00     -•19694*85-00     -.501^^836-00
   -.817*8^2"-00      -.1^^433*^+01     -•30254090+01     -•12321^13+02     -.458^7690+02

OUTPUT FROM FESI

     .20820^3^-00      .170449^6-0^      •11752"0"+00      •60668^71-01      .255"7574-01
     .9027*^9^-02      .27*791**-0^      •72575328-03      •17061"04-03      .359^^171-04

OUTPUT FROM FESK

     .612*7^9*-00      .685765^8-0^      •9^89291^-00      •16549"88+01      .346^^985+01
     .86649,4^+01      .2^*61^**+0^      •8^5640"^+02      •*2803^45+03      .141^7495+04
     Y=    .*9^7678+01       NU=    .000^0^0          N=    -9
```

continued

```
OUTPUT FROM BESJ

   -.39960906-00    .46797095-01    .37343720-00    -.42823841-00    .27408947-00
   -.12701959-00    .46553604-01   -.14245722-01     .73364321-02   -.86027088-03

OUTPUT FROM BESY

    .15872660-06   -.46058470-00    .20342351-00     .19694785-00   -.50100836-00
    .81778020-00   -.16647374+01    .30254000+01    -.12321513+02    .45867690+02

OUTPUT FROM BESI

    .20820635-00    .17044966-00    .11752404+00     .40668071-01    .25507573-01
    .90273311-02    .22779001-02    .73577211-03     .17054178-03    .36310142-04

OUTPUT FROM BESK

    .61277097-00    .61576518-00    .95892918-00     .16549488+01    .34678985+01
    .86609486+01    .25761673+02    .85564045+02     .32803045+03    .14117495+04
    X=   .3015587+01      NU=   .0000000         N=     0


OUTPUT FROM BESJ

    .30011576-00    .12076592-04   -.30011577-00    -.17111380-00    .15377262-00
    .34646340-00    .34007555-00    .27522791-00     .12933501-00    .59778169-01

OUTPUT FROM BESY

   -.21234346-01   -.30236207-00   -.64962880-01     .26532090-00    .29187723-00
    .67509657-01   -.19564672-00   -.40216258-00    -.40688042-00   -.98107124-00

OUTPUT FROM BESI

    .15356000-00    .14015140-00    .11303555+00     .77703167-01    .46500811-01
    .24586031-01    .11535641-01    .40547491-02     .18477350-02    .64037830-03

OUTPUT FROM BESK

    .46574356-00    .49746357-00    .60716025-00     .84364144-00    .13286749+01
    .23587632+01    .46008358+01    .10382320+02     .35409340+02    .68331774+02
    X=   .3015587+01      NU=   .0000000         N=    -3


OUTPUT FROM BESJ

    .30011576-00   -.12076592-04   -.30011577-00     .17111380-00    .15377262-00
   -.34646340-00    .34007555-00   -.27522791-00     .12933502-00   -.59778182-01

OUTPUT FROM BESY

   -.21234346-01    .30236207-00   -.64962880-01    -.26532090-00    .29187723-00
   -.67509657-01   -.19564672-00    .40216258-00    -.40688042-00    .98107124-00

OUTPUT FROM BESI

    .15356000-00    .14015140-00    .11303555+00     .77703167-01    .46500810-01
    .24586230-01    .11535657-01    .40547566-02     .18477162-02    .64078875-03

OUTPUT FROM BESK

    .46574356-00    .49746357-00    .60716025-00     .84364144-00    .13286749+01
    .23587632+01    .46008358+01    .10382320+02     .35409340+02    .68331774+02

EXECUTION TERMINATED BY AN ATTEMPT TO READ THE END OF-DATA.
ABOVE FAULT AT INTERNAL SEQUENCE NUMBER    00105  IN  TES-
```

SAMPLE INPUT (FOR MATAB)

Test programs for real matrix addition and subtraction.

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEFFM
- PLT,A MATAB
- FOR TEST3
C     MAIN PROGRAM TO TEST 1107 MATRIX PACKAGE - REAL MATRIX ADDITION
C     AND SUBTRACTION.  SET J=1 FOR ADDITION J=0 FOR SUBTRACTION.
C
      DIMENSION A(64),B(64),C(64)
C
C     PROGRAM INITIALIZATION.
C
      WRITE (6,100)
    1 READ (5,101) J,L,M,N
      N1 = L*M
      READ (5,102) (A(I),I=1,N1)
      READ (5,102) (B(I),I=1,N1)
C
      IF (J.EQ.0) GO TO 2
C
      CALL MXA(A,B,C,M,N)
      GO TO 3
C
    2 CALL MXS(A,B,C,M,N)
      GO TO 4
C
C     OUTPUT.
C
    3 WRITE (6,103) (C(I),I=1,N1)
      GO TO 1
    4 WRITE (6,104) (C(I),I=1,N1)
C
C     FORMAT STATEMENTS.
C
  100 FORMAT (59H1MAIN PROGRAM TO TEST REAL MATRIX ADDITION AND SUBTRACT
     1ION.////)
  101 FORMAT (4I5)
  102 FORMAT(F10.0)
  103 FORMAT(//36H MATRIX C FOR REAL ADDITION ROUTINE.//(1H E20.8))
  104 FORMAT(//39H MATRIX C FOR REAL SUBTRACTION ROUTINE.//(1H E20.8))
      END
- XQT TEST3
    1    3    3    3
        2.
        3.
        4.
        1.
        4.
        1.
        2.
        1.
        4.
        1.
        2.
        0.
        3.
        7.
        2.
        1.
        8.
        5.
    0    3    3    3
        2.
        3.
        4.
        1.
        4.
        1.
        2.
        1.
```

```
          4.
          1.
          2.
          0.
          3.
          7.
          2.
          1.
          8.
          5.
- FIN
```

SAMPLE OUTPUT (FROM MATAB)

Test program for real matrix addition and subtraction.

```
MAIN PROGRAM TO TEST REAL MATRIX ADDITION AND SUBTRACTION.



MATRIX C FOR REAL ADDITION ROUTINE.

        .30000000+01
        .50000000+01
        .40000000+01
        .40000000+01
        .11000000+02
        .30000000+01
        .30000000+01
        .90000000+01
        .90000000+01

MATRIX C FOR REAL SUBTRACTION ROUTINE.

        .10000000+01
        .10000000+01
        .40000000+01
       -.20000000+01
       -.30000000+01
       -.10000000+01
        .10000000+01
       -.70000000+01
       -.10000000+01
```

SAMPLE INPUT (FOR MATAB)

Test program for real matrix multiplication.

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEEFM
- PLT,A MATAB
- FOR TEST4
C     MAIN PROGRAM TO TEST 1107 MATRIX PACKAGE - REAL MATRIX MULTIPLICAT
C     ION.
C
      DIMENSION A(3,3),B(3,3),C(3,3)
C
C     PROGRAM INITIALIZATION.
C
      WRITE (6,100)
    1 READ (5,101) M, N, L
      READ (5,102) ((A(I,J),J=1,L),I=1,M)
      READ (5,102) ((B(I,J),J=1,N),I=1,L)
C
      CALL MXM(A,B,C,M,N,L)
C
C     OUTPUT.
C
      WRITE (6,103) ((I,J,C(I,J),J=1,N),I=1,M)
      GO TO 1
C
C     FORMAT STATEMENTS.
C
  100 FORMAT(49H1MAIN PROGRAM TO TEST REAL MATRIX MULTIPLICATION.////)
  101 FORMAT(3I5)
  102 FORMAT(F10.0)
  103 FORMAT(4X2H I,3X2H J,8X7H C(I,J)//(1H 2I5,E20.8))
      END
- XQT TEST4
    3    3    3
         3.
         2.
         2.
         4.
         3.
         0.
         5.
         0.
         5.
         1.
         1.
         1.
         1.
         1.
         1.
         1.
         1.
         1.
    3    3    3
         3.
         2.
         2.
         4.
         3.
         0.
         5.
         0.
         5.
         2.
         2.
         2.
         2.
         2.
         2.
         2.
         2.
         2.
- FIN
```

SAMPLE OUTPUT (FOR MATAB)

Test program for real matrix multiplication.

```
MAIN PROGRAM TO TEST REAL MATRIX MULTIPLICATION.


   I    J          C(I,J)

   1    1         .70000000+01
   1    2         .70000000+01
   1    3         .70000000+01
   2    1         .10000000+01
   2    2         .10000000+01
   2    3         .10000000+01
   3    1         .10000000+02
   3    2         .10000000+02
   3    3         .10000000+02
   I    J          C(I,J)

   1    1         .14000000+02
   1    2         .14000000+02
   1    3         .14000000+02
   2    1         .20000000+01
   2    2         .20000000+01
   2    3         .20000000+01
   3    1         .20000000+02
   3    2         .20000000+02
   3    3         .20000000+02

EXECUTION TERMINATED BY AN ATTEMPT TO READ THRU END-OF-DATA.
ABOVE FAULT AT INTERNAL SEQUENCE NUMBER    00105  IN  TEST4
```

SAMPLE INPUT (FOR MATAB)

Test program for complex matrix postmultiplication.

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEFFM
- PLT,A MATA3
- FOR TEST
C     MAIN PORGRAM TO TEST 1107 MATRIX PACKAGE - COMPLEX MATRIX POSTMULT
C     IPLICATION.
C
      DIMENSION A(2,3,3), B(2,3), C(2,3,3)
C
C     PROGRAM INITIALIZATION.
C
      WRITE (6,100)
    1 READ (5,101) M, N
      READ (5,102) ((A(1,I,J),A(2,I,J),J=1,N),I=1,M)
      READ (5,102) (B(1,I),B(2,I),I=1,N)
C
      CALL CMXMD(A,B,C,M,N)
C
C     OUTPUT.
      WRITE (6,103) ((I,J,C(1,I,J),C(2,I,J),J=1,N),I=1,M)
      GO TO 1
C
C     FORMAT STATEMENTS.
C
  100 FORMAT(56H1MAIN PROGRAM TO TEST COMPLEX MATRIX POSTMULTIPLICATION.
     1////)
  101 FORMAT(2I5)
  102 FORMAT(2F10.0)
  103 FORMAT (25H POSTMULTIPLICATION OF A.//4X2H I,3X2H J,18X7H A(I,J)//
     1 (1H 2I5,2E20.8))
      END
- XQT TEST
    3    3
         2.        1.
         2.        1.
         3.        1.
         4.        1.
         5.        1.
         6.        1.
         7.        1.
         8.        1.
         9.        1.
         2.        1.
         2.        1.
         2.        1.
    3    3
         1.        1.
         2.        2.
         3.        3.
         4.        4.
         5.        5.
         6.        6.
         7.        7.
         8.        8.
         9.        9.
         2.        2.
         1.        1.
         0.        0.
- FIN
```

Q5

SAMPLE OUTPUT (FROM MATAB)

Test program for complex matrix postmultiplication.

```
 MAIN PROGRAM TO TEST COMPLEX MATRIX POSTMULTIPLICATION.


 POSTMULTIPLICATION OF A.

     I    J                    A(I,J)

     1    1         .30000000+01          .40000000+01
     1    2         .30000000+01          .40000000+01
     1    3         .50000000+01          .50000000+01
     2    1         .70000000+01          .60000000+01
     2    2         .90000000+01          .70000000+01
     2    3         .11000000+02          .80000000+01
     3    1         .13000000+02          .90000000+01
     3    2         .15000000+02          .10000000+02
     3    3         .14999999+02          .11000000+02
 POSTMULTIPLICATION OF A.

     I    J                    A(I,J)

     1    1        -.00000000             .40000000+01
     1    2        -.00000000             .40000000+01
     1    3        -.00000000             .00000000
     2    1        -.00000000             .16000000+02
     2    2        -.00000000             .10000000+02
     2    3        -.00000000             .00000000
     3    1        -.00000000             .27999999+02
     3    2        -.00000000             .16000000+02
     3    3        -.00000000             .00000000


EXECUTION TERMINATED BY AN ATTEMPT TO READ THRU END-OF-DATA.
ABOVE FAULT AT INTERNAL SEQUENCE NUMBER    00105  IN  TES-
```

SAMPLE INPUT (FOR MATAB)

Test program for complex matrix transpose.

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=BEEF M
- PLT,A MAT/ 3
- FOR TEST
C     MAIN PROGRAM TO TEST 1107 MATRIX PACKAGE - COMPLEX MATRIX TRANSPOS
C     E.
C
      DIMENSION A(2,5,3),C(2,3,5)
C
C     PROGRAM INITIALIZATION.
C
      WRITF (6,100)
    1 READ (5,101) M, N
      READ (5,102) ((A(1,I,J),A(2,I,J),J=1,N),I=1,M)
C
    - CALI. CMXT(A,C,M,N)
C
C     OUTPUT.
C
      WRITE (6,103) ((I,J,C(1,I,J),C(2,I,J),J=1,N),I=1,M)
      GO TO 1
C
C     FORMAT STATEMENTS.
C
  100 FORMAT (47H1MAIN PROGRAM TO TEST COMPLEX MATRIX TRANSPOSE.////)
  101 FORMAT (2I5)
  102 FORMAT (2F10.0)
  103 FORMAT (31H COMPLEX MATRIX TRANSPOSE OF A.//4X2H I,3X2H J, 19X7H A
     1(I,J)//(1H 2I5,2E20.8))
      END
- XQT TEST
    5    3
        6.        8.
        4.       -2.
        1.        0.
        0.       10.
        6.       -2.
        4.       -4.
        2.       -8.
        8.        3.
        4.       -2.
        3.        2.
        4.        6.
        2.       -8.
        3.       -2.
        2.       -1.
        1.        0.
- FIN
```

SAMPLE OUTPUT (FROM MATAB)

Test program for complex matrix transpose.

```
MAIN PROGRAM TO TEST COMPLEX MATRIX TRANSPOSE.


COMPLEX MATRIX TRANSPOSE OF A.

     I    J                 A(I,J)

     1    1        .60000000+01         .80000000+01
     1    2        .00000000            .10000000+02
     1    3        .20000000+01        -.80000000+01
     2    1        .40000000+01        -.20000000+01
     2    2        .60000000+01        -.20000000+01
     2    3        .80000000+01         .30000000+01
     3    1        .10000000+01         .00000000
     3    2       -.40000000+01        -.40000000+01
     3    3       -.40000000+01        -.20000000+01
     4    1        .00000000            .10000000+02
     4    2        .20000000+01        -.80000000+01
     4    3        .30000000+01         .20000000+01
     5    1        .60000000+01        -.20000000+01
     5    2        .80000000+01         .30000000+01
     5    3       -.40000000+01         .60000000+01


EXECUTION TERMINATED BY AN ATTEMPT TO READ THRU END-OF-DATA.
ABOVE FAULT AT INTERNAL SEQUENCE NUMBER    00105  IN  TEST
```

SAMPLE INPUT (FOR CINV1)

```
- RUN HIDOLL,ZJ1190,3,10
- ASG A=REFFM
- PLT,A INV,CINV1
- FOR TEST2
C      MAIN PROGRAM TO TEST DETERMINANT EVALUATION AND MATRIX INVERSION
C      OF COMPLEX AND REAL MATRICES.  SET IER=0 FOR COMPLEX MATRIX, IER=1
C      FOR REAL MATRIX.  DIVIDE CHECK USED FOR SINGULARITY.
C
       DIMENSION AREAL(50),BIMAG(50),K(1),IK(1)
       EQUIVALENCE (AREAL(1),K(1)),(AREAL(2),K(2)),(AREAL(3),K(3)),(BIMAG
      1(1),IK(1)),(BIMAG(2),IK(2)),(BIMAG(3),IK(3))
C
C      PROGRAM INITIALIZATION.
C
       WRITE (6,100)
     1 READ (5,101) N, IER
       KN=N*N
       M=KN+3
       IF (IER.EQ.0) GO TO 2
       READ (5,102) (AREAL(L), L=M,4,-1)
       K(2)=N
       K(3)=N
       GO TO 5
     2 READ (5,103) (AREAL(L),BIMAG(L), L=M,4,-1)
       IK(2)=N
       IK(3)=N
C
C      CALL SEQUENCE.
C
       CALL CINV1 (AREAL,BIMAG,IER)
       IF (IER.EQ.1) GO TO 7
       IF (IER.EQ.-1) GO TO 8
       GO TO 10
     5 DO 6 L=1,M
       BIMAG(L)=0.0
     6 CONTINUE
       CALL CINV1(AREAL,BIMAG,IER)
       IF (IER.EQ.1) GO TO 7
       IF (IER.EQ.-1) GO TO 8
       GO TO 11
C
C      OUTPUT.
C
     7 WRITE (6,104)
       GO TO 1
     8 WRITE (6,105)
       GO TO 1
    10 WRITE (6,106) AREAL(1),BIMAG(1)
       WRITE (6,107) (AREAL(L),BIMAG(L), L=M,4,-1)
       GO TO 1
    11 WRITE (6,108) AREAL(1)
       WRITE (6,109) (AREAL(L), L= M,4,-1)
       GO TO 1
C
C      FORMAT STATEMENTS.
C
   100 FORMAT(96H1MAIN PROGRAM TO TEST INVERSION AND DETERMINANT EVALUATI
      1ON OF GENERAL COMPLEX AND REAL MATRICES.////)
   101 FORMAT(2I5)
   102 FORMAT(F10.0)
   103 FORMAT(F10.0,F10.0)
```

continued

```
   104 FORMAT(20H MATRIX IS SINGULAR.//)
   105 FORMAT(31H MATRIX NOT IN ACCEPTABLE FORM.//)
   106 FORMAT(15H DETERMINANT = 2E20.8//)
   107 FORMAT(16H INVERSE MATRIX.//(1H 2E20.8))
   108 FORMAT(15H DETERMINANT = E20.8//)
   109 FORMAT(16H INVERSE MATRIX.//(1H E20.8))
       END
- XQT TEST2
    3    1
   9.3746
   3.0416
   2.4371
   3.0416
   6.1832
   1.2163
   2.4371
   1.2163
   8.4429
   3    0
   9.3746    9.3746
   3.0416    3.0416
   2.4371   -2.4371
   3.0416    3.0416
   6.1832    6.1832
   1.2163    1.2163
   2.4371   -2.4371
   1.2163    1.2163
   3.4429    3.4429
- FIN
```

SAMPLE OUTPUT (FROM CINV1)

```
MAIN PROGRAM TO TEST INVERSION AND DETERMINANT EVALUATION OF GENERAL COMPLEX AND REAL MATRICES.


DETERMINANT =          .34265028+03


INVERSE MATRIX.

        .14803262-00
       -.83503731-01
        .54773283-01
       -.83503731-01
        .21365058-00
       -.54908796-01
        .54773283-01
       -.54908796-01
        .14216365-00
DETERMINANT =          -.10818158+03          .10818158+03


INVERSE MATRIX.

        .90952539-01        -.90952539-01
       -.67707260-01         .67707260-01
        .94703922-01        -.94703922-01
       -.67707260-01         .67707260-01
        .13289001-00        -.13289001-00
       -.94938225-01         .94938225-01
        .94703922-01        -.94703922-01
       -.94938225-01         .94938225-01
        .24580333-00        -.24580333-00

EXECUTION TERMINATED BY AN ATTEMPT TO READ THRU END-OF-DATA.
ABOVE FAULT AT INTERNAL SEQUENCE NUMBER    00106  IN  TEST2
```

COST ESTIMATE

For the jobs listed on the Sample Input, the processing times are listed below. The current rate for the University of Notre Dame is $480./hour for computer time with an additional charge of $10. for postage and handling.

Charge to user = computer costs + postage and handling + network overhead

Program	*Processing Time (in seconds)*	*Computer costs*	*Charge to user* *
GAMMA	16	$2.51	$12.51
BES	23	$3.74	$13.74
MATAB (add. & sub.)	32	$4.67	$14.67
MATAB (multiplic.)	43	$6.11	$16.11
MATAB (postmultip.)	29	$4.23	$14.23
MATAB (transpose)	32	$4.74	$14.74
CINV1	32	$4.82	$14.82

* A figure for network overhead must still be added to the totals in this column.

CONTENTS—BEEFM

pages

1 Identification & Abstract
3–11 UserUser Instructions
13–14 I/O (GAMMA)
15–20 I/O (BES)
21–30 I/O (MATAB)
31–33 I/O (CINV1)
35 Cost—Contents

DESCRIPTIVE TITLE	Bio-Medical Multivariate Statistical Programs
CALLING NAME	BMD
INSTALLATION NAME	University of Notre Dame Computing Center
AUTHOR(S) AND AFFILIATION(S)	School of Medicine University of California, Los Angeles UNIVAC Division of Sperry Rand Corporation College of Business Administration and Computing Center University of Notre Dame
LANGUAGE	FORTRAN IV
COMPUTER	UNIVAC 1107
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Elizabeth Hutcheson, EIN Technical Representative, Computing Center, University of Notre Dame, Notre Dame, Ind. 46556 Tel.: (219) 283-7784

FUNCTIONAL ABSTRACT

The BMD system is a package of computer programs designed to do both basic data processing and the subsequent statistical analysis. The programs have been prepared in an easy-to-use parametric form so that the researcher may adapt them to a wide variety of statistical problems. For further details of the package, see the User Instructions. The BMD is available at the University of Notre Dame in its 1967 edition.

REFERENCES

Dixon, W.J. (Ed.), *BMD: Biomedical Computer Programs*, (Univ. of Calif. Press, Berkeley, 1967, 1970).

Univ. of Notre Dame Computing Center, *BMD for the UNIVAC 1107*, (Rough draft, 1967 ed. of BMD). Available from Univ. of Notre Dame Computing Center, Notre Dame, Ind.

USER INSTRUCTIONS

The computer routines which constitute the BMD multivariate statistical package are designed for use by statisticians and researchers; their use does not require the services of a computer programmer. The user of the statistical system enters his data on punched cards in an easy-to-use and flexible format. A problem header card is also punched which defines the size and options to be used for each analysis to be performed. Detailed instructions for the preparation of these cards is included in Ref. 1 and 2.

The card decks are then forwarded to the UNIVAC 1107 or 1108 computer installation where system cards are added These systems cards call the appropriate computer program and define any tape files used. The analysis is then performed on the computer and the print-outs sent back to the researcher. Many computer installations are able to perform this service (even for large computations) for researchers within a day.

Keypunching of the basic input can, however, often be a detailed and time consuming operation. Researchers should contact the keypunching department before they start accumulating their raw data so that it can be collected in a format lending itself to rapid, accurate transcription.

The following programs comprise the BMD package:

Name	*Description*
BMD01D	SIMPLE DATA DESCRIPTION This program computes simple averages and measures of dispersion of variables, omitting those values which the user specifies for exclusion from the computations.
BMD02D	CORRELATION WITH TRANSGENERATION This program computes simple correlation coefficients, averages and measures of dispersion on entering variables and/or transgenerated variables from selected cases whose values for specified variables have a precise logical relationship in agreement with a specified Boolean expression. Cross-tabulation plots are optional.
BMD03D	CORRELATION WITH ITEM DELETION This program computes a simple correlation matrix,

continued

Name	*Description*
	omitting values of variables that the user specifies to be deleted. Blanks as well as other codes may be specified for deletion, e.g., to indicate missing values.
BMD04D	ALPHANUMERIC FREQUENCY COUNT (1 COLUMN DATA) This program computes frequencies of legal characters on one-column data. Any legal numeric, alphabetic, or special character is counted. The input data may be on cards, or on magnetic tape in Field Data mode.
BMD05D	GENERAL PLOT INCLUDING HISTOGRAMS This program provides a method by which graphs and histograms can be produced.
BMD06D	DESCRIPTION OF STRATA Cases are separated into groups based on specified intervals of one variable, the conditioning variable. For these selected groups, computations are performed for any other variables which are designated as the conditioned variables. Univariate descriptions and correlation coefficients are included.
BMD07D	DESCRIPTION OF STRATA WITH HISTOGRAMS This program groups the data into a specified number of groups based on the order of entry of the data or into groups whose values for a base variable are within intervals established by specified cut points. For these groups, histograms are printed for each variable. The number of classes or categories of the histograms may be specified or they may be computed by the program. Univariate descriptions and correlation coefficients are included.
BMD08D	CROSS TABULATION WITH VARIABLE STACKING This program computes two-way frequency tables of data input. Frequency tables are computed from specified ranges of the original variables, variables after transgeneration, stacked variables or combinations of these. Data input may be positive or negative integers only.
BMD09D	CROSS TABULATION, INCOMPLETE DATA This program performs cross-tabulations of input data,

continued

Name	*Description*
	excluding from cross-tabulation specified special values or codes used to designate missing values. The program differentiates between blanks and zeros in testing for special values.
BMD10D	DATA PATTERNS FOR DICHOTOMIES This program finds frequencies and patterns of any one particular specified code in the input data. Frequent use will be for a code representing missing values.
BMD11D	DATA PATTERNS FOR POLYCHOTOMIES This program prints patterns of one-column data and item numbers or case numbers to identify cases having these data patterns. If desired, original data may be recoded by the program before the data patterns are printed.
BMD01M	PRINCIPAL COMPONENT ANALYSIS This program computes the principal components of standardized data and rank-orders each standardized case by the size of each principal component separately.
BMD02M	REGRESSION ON PRINCIPAL COMPONENTS This program computes the principal components of standardized data and rank-orders each standardized case by the size of each principal component separately. Each dependent variable is regressed on the first, first two, first three and all principal components when each component is expressed in terms of standardized data.
BMD03M	GENERAL FACTOR ANALYSIS This program performs a principal component solution and an orthogonal rotation of the factor matrix. Data input to this program may be in the form of raw data, a correlation matrix, or a factor matrix. Data input may be read in from punched cards, Field Data or binary tape.
BMD04M	DISCRIMINANT ANALYSIS FOR TWO GROUPS This program computes a linear function of p variables measured on each individual of two groups. This function can serve as an index for discrimination between the groups. It is determined from

continued

Name	*Description*
	from the criterion of "best" in that the difference between the mean indices for the two groups divided by a pooled standard deviation of the indices is maximized.
BMD05M	DISCRIMINANT ANALYSIS FOR SEVERAL GROUPS This program directs the computation of a set of linear functions for the purpose of classifying an individual into one of several groups. The input data consist of a set of observations for each of the classification groups; each observation consists of the values of a set of variables, and each observation contains a value for each of the variables.
BMD06M	CANONICAL ANALYSIS This program computes the canonical correlations between two sets of variables.
BMD07M	STEPWISE DISCRIMINANT ANALYSIS This program performs a multiple discriminant analysis in a stepwise manner. At each step one variable is entered into the set of discriminating variables. The entering variable is selected by the following equivalent criteria; the variable with the largest F value (see computational procedure), or the variable which when partialed on the previously entered variables has the highest multiple correlation with the groups, or the variable which gives the greatest decrease in the ratio of within to total generalized variances. A variable is deleted if its F value becomes too low. The program also computes canonical correlations and coefficients for canonical variables. It plots the first two canonical variables to give an optimal two-dimensional picture of the dispersion.
BMD01R	SIMPLE LINEAR REGRESSION This program performs simple linear regression analysis on a single or combined treatment groups with unequal sample sizes. (The words "treatments groups" are used here to describe categories.) The "within" cross-products sums and coefficients are computed; thus, analysis-of-covariance information is also provided in the output.

continued

Names	*Description*
BMD02R	STEPWISE REGRESSION This program computes a sequence of multiple linear regression equations in a stepwise manner. At each step one variable is added to the regression equation. The variable added is the one which makes the greatest reduction in the error sum of squares. Equivalently, it is the variable which has highest partial correlation with the dependent variable partialed on the variables which have already been added; and equivalently, it is the variable which, if it were added, would have the highest F value. In addition, variables can be forced into the regression equation. Non-forced variables are automatically removed when their F values become too low. Regression equations with or without the regression intercept may be selected.
BMD03R	MULTIPLE REGRESSION WITH CASE COMBINATIONS This program performs multiple regression and correlation analysis on the data within various combinations of subsamples from the same population.
BMD04R	PERIODIC REGRESSION AND HARMONIC ANALYSIS This program performs periodic or harmonic regression analysis using the regression function of the Fourier series form. Periodic regressions are computed successively up to the harmonic specified by the user.
BMD05R	POLYNOMIAL REGRESSION This program computes polynomial regressions of the form $Y = \alpha + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k$ (k^{th} degree), where k is some positive integer.
BMD06R	ASYMPTOTIC REGRESSION This program performs asymptotic regression analysis using the modified exponential funtion of the form, $Y = \alpha + \beta_\mu x.$ By making a transgeneration (or successive transgenerations) on the dependent variable Y, other equations can be used.

continued

Name	*Description*
BMD01S	LIFE TABLE AND SURVIVAL RATE This program computes proportions surviving, survival rates and standard errors for successively reduced time periods and prints a plot of the cumulative proportion surviving.
BMD02S	CONTINGENCY TABLE ANALYSIS This program computes two-way frequency and percentage tables, chi squares, contingency coefficients, and maximum likelihood ratios. Each variable may be categorized in several different ways; each way is referred to as a categorization.
BMD03S	BIOLOGICAL ASSAY; PROBIT ANALYSIS This program obtains maximum likelihood estimates for the parameters α and β in the probit equation $y = \alpha + \beta x$. An iterative scheme is used. As an option, an estimate of the natural (threshold) response rate is also obtained.
BMD04S	GUTTMAN SCALE PREPROCESSOR This program performs the initial step of BMD06S and BMD07S (GUTTMAN SCALE NO. 2, PARTS 1 AND 2). It weights the input data, checks the data, determines the frequencies of responses, combines those responses which have frequencies less than N% (N is user-specified) of the total respondents, ranks the respondents by the Cornell technique, and determines the errors.
BMD05S	GUTTMAN SCALE NO. 1 This program takes the data on many variables, assigns proper weights to the data, ranks the respondents (or cases) from most favorable to least favorable, and assigns a Guttman scale score for each respondent. This procedure is widely used for data from attitude or opinion questionnaires.
BMD06S	GUTTMAN SCALE NO. 2, PART 1 This program performs the initial steps for GUTTMAN SCALE NO. 2. The Problem, Response, and Reflection (if any) Cards are read and processed. The input data are read into the computer, weighted, and checked for valid values.

continued

Name	*Description*
BMD07S	GUTTMAN SCALE NO. 2, PART 2 This program performs the major computations of GUTTMAN SCALE NO. 2. It reads and processes any Forced Combination Cards and performs any and all combinations of responses which will give the best possible Guttman scale the program can produce. During the running of the program, the respondents are ranked by the Cornell technique after each combination and the errors to each response are determined.
BMD08S	GUTTMAN SCALE NO. 2, PART 3 This program performs the final steps of GUTTMAN SCALE NO. 2. In this program, weights are assigned to all zero responses such that no additional errors are introduced. Guttman scale scores are assigned, and the final results specified by the user are printed out.
BMD09S	TRANSGENERATION This program performs selected transgenerations on specified variables in the data. Any of the transgeneration codes listed in the BMD manual may be selected. Input may be from punched cards, from Field Data tape, or from binary tape.
BMD10S	TRANSPOSTION OF LARGE MATRICES This program performs no analysis but is designed to convert large matrices to the form required by some programs which do perform statistical analysis (e.g., to convert data from "case-wise" to "variable-wise" form for use in BMD01V). Large matrices are read in from cards or Field Data tape, transposed, and written out on Field Data tape.
BMD01T	AMPLITUDE AND PHASE ANALYSIS This program computes the amplitude and phase of moderately wide-band noise and noise contaminated by extraneous noise. The amplitude and phase are determined from a pair of finite moving averages on the sample noise. A generalized Tukey filter with a variable number of triangles and resolutions is used.

continued

Name	*Description*
BMD02T	AUTOCOVARIANCE AND POWER SPECTRAL ANALYSIS This program computes the autocovariance, power spectrum, cross covariance, cross spectrum, transfer function, and coherence function of time series.
BMD01V	ANALYSIS OF VARIANCE FOR ONE-WAY DESIGN This program computes an analysis-of-variance table for one variable of classification, with unequal group sample sizes.
BMD02V	ANALYSIS OF VARIANCE FOR FACTORIAL DESIGN This program computes an analysis of variance for a factorial design.
BMD03V	ANALYSIS OF COVARIANCE FOR FACTORIAL DESIGN This program computes a full factorial analysis of covariance.
BMD04V	ANALYSIS OF COVARIANCE WITH MULTIPLE COVARIATES This program is designed to compute analysis-of-covariance information for one analysis-of-variance variable with multiple covariates and unequal treatment group sizes. Cases may be specified by the user as being in certain treatment groups, or cases may be placed in treatment groups by the program in accordance with a specified Boolean expression.
BMD05V	GENERAL LINEAR HYPOTHESIS This program performs the calculations required for a general linear hypothesis model. The independent variables are of two general types, variables used to specify the analysis-of-variance classification, and variables used as covariates. By use of these variables, the program can be used for balanced or unbalanced analysis-of-variance or covariance designs and missing-value problems.
BMD06V	GENERAL LINEAR HYPOTHESIS WITH CONTRASTS This program is similar to BMD05V in that it is designed to estimate and test the statistical significance of the parameters which occur in the general linear hypothesis model. This program is more general in that it can test the statistical significance of any real valued linear function

continued

Name	*Description*
	of the parameters. It is less general in that higher order simultaneous tests cannot be made. Thus, this program can test the hypothesis $\beta_1 = \beta_2 = 0$.
BMD07V	MULTIPLE RANGE TESTS This program computes an analysis-of-variance table for one variable of classification, with unequal group sizes. The treatment group or category means are ranked in increasing order; and a multiple range test, using significant ranges input by the user, is performed on the ranked means. A maximum of 9 range tests may be done on the ranked means.
BMD08V	ANALYSIS OF VARIANCE This program performs analysis of variance for any hierarchial design with equal cell sizes. This includes the nested, partially nested and partially crossed, and fully crossed designs. The model is specified by indicating the nesting relationship of the indices. Separate analyses may be performed on several dependent variables simultaneously. Each analysis of variance table includes an expected mean square in terms of the population variance components. For this calculation each index may be specified as fixed or random from a finite or infinite population.

REFERENCES

Dixon, W.J. (Ed.), *BMD: Biomedical Computer Programs*, (Univ. of Calif. Press, Berkeley, 1967, 1970).

Univ. of Notre Dame Computing Center, *BMD for the UNIVAC 1107*, (Rough draft, 1967 ed. of BMD). Available from Univ. of Notre Dame Computing Center, Notre Dame, Ind.

Anderson, T.W., *Introduction to Multivariate Statistical Analysis*, (John Wiley & Sons, Inc., New York, 1958). This text surveys most of the procedures used by BMD Programs.

Cooley, W.W. and Lohnes, P.R., *Multivariate Procedures for the Behavioral Sciences*, (John Wiley and Sons, Inc., New York, 1962). The MIT statistical package is described in this text.

continued

REFERENCES

Dixon, W.J., and Massey, F.J., *Introduction to Statistical Analysis*, (McGraw-Hill Book Co., Inc., New York, 1957), 2nd ed. Statistical definitions, simpler standard statistical procedures, and reference tables for tests and estimation techniques are included.

Kendall, M.G., *A Course in Multivariate Analysis*, (Hafner Pub. Co., New York, 1957), Ch. I. The first chapter of this text gives an introduction to the scope and problems considered by multivariate statistical analysis.

SAMPLE INPUT AND OUTPUT

See Dixon, W.J. (Ed.), *BMD: Biomedical Computer Programs*, (Univ. of Calif. Press, Berkeley, 1967).

COST ESTIMATE

See references listed in USER INSTRUCTIONS.

Charge to user = computer time + postage and handling + network overhead
= computer time + $10.00 + network overhead

CONTENTS—BMD

pages
1 Identification & Abstract
3–12 User Instructions
13 I/O
15 Cost—Contents

DESCRIPTIVE TITLE	UNIVAC 1107 Linear Programming Package
CALLING NAME	LP1107
INSTALLATION NAME	The University of Notre Dame Computing Center
AUTHOR(S) AND AFFILIATION(S)	UNIVAC Division of Sperry Rand
LANGUAGE	Sleuth II/FORTRAN IV
COMPUTER	UNIVAC 1107
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Elizabeth Hutcheson, EIN Technical Representative, Computing Center, Univ. of Notre Dame, Notre Dame, Ind. 46556 Tel.: (219) 283-7784

FUNCTIONAL ABSTRACT

LP1107 is a generalized program for the solution of linear programming problems. More specifically, the LP1107 command structure is a mathematical programming control language; used to prepare a control sequence for a specific programming job.

Either the dual or simplex algorithm, or both, can be used. LP1107 incorporates a true programming language with logical capabilities and full macro capabilities. If a group of commands is being used repetitively, the user may incorporate it into his macro command library and issue the macro command instead. Single and double precision arithmetic are included.

REFERENCES

UNIVAC Systems Programming Library Services, *UP3897 1107LP User's Manual (preliminary)*, (UNIVAC, Sperry Rand Building, New York, 1964).

Aronofsky, J.S., "Growing Applications of Linear Programming with a View Towards Understanding Principles on Which Future LP Systems Must be Based," *Communications of the ACM*, (June 1964).

Copies of these references are available through the local UNIVAC representatives or from the EIN Office at the cost of reproduction and mailing.

USER INSTRUCTIONS

The user is directed to the references for a full description of the operating system and programming language.

REFERENCES

UNIVAC Systems Programming Library Services, *UP3897 1107LP User's Manual (preliminary)*, (UNIVAC, Sperry Rand Building, New York, 1964).

Aronofsky, J.S., "Growing Applications of Linear Programming with a View Towards Understanding Principles on Which Future LP Systems Must be Based," *Communications of the ACM*, (June 1964).

Copies of these references are available through the local UNIVAC representatives or from the EIN Office at the cost of reproduction and mailing.

SAMPLE INPUT

The problem shown is that described in *Introduction to Linear Programming* by Charnes, Cooper and Henderson, (John Wiley & Sons, New York, 1953).

```
∇ RUN SMITH, 077707,5,20
∇ XQT CUR
TRW F
IN F
TRW F
∇  XQT 1107LP
SET CASE1 TO (NUT-MI)  ; DEFINE CONTENTS OF CASE1
SET CASE2 TO (X     )  ; DEFINE CONTENTS OF CASE2
TITLE                  ; NEXT CARD CONTAINS PAGE HEADING
 1107LP RUN ON NUT-MIX PROBLEM
LOAD                   ; LOAD SHARE FORMAT MATRIX
ROW ID
     1    +MA-CAS
     1    +MA+PEA
     1    +MB-CAS
     1    +MB+PEA
     1    +XX+CAS
     1    +XX+PEA
     1    +XX+HAZ
MATRIX
            XX+CAS  1.0
            XX+PEA  1.0
            XX+HAZ  0.6
      MA/CAS$$$$$$  0.5
      MA/CASMA-CAS -0.5
      MA/CASMA+PEA -0.25
      MA/CASXX+CAS  1.0
      MA/PEA$$$$$$  0.5
      MA/PEAMA-CAS  0.5
      MA/PEAMA+PEA  0.75
      MA/PEAXX+PEA  1.0
      MA/HAZ$$$$$$  0.5
      MA/HAZMA-CAS  0.5
      MA/HAZMA+PEA -0.25
      MA/HAZXX+HAZ  1.0
      MB/CAS$$$$$$  0.35
      MB/CASMB-CAS -0.75
      MB/CASMB+PEA -0.5
      MB/CASXX+CAS  1.0
```

continued

```
      MB/PEA$$$$$$  0.35
      MB/PEAMB-CAS  0.25
      MB/PEAMB+PEA  0.5
      MB/PEAXX+PEA  1.0
      MB/HAZ$$$$$$  0.35
      MB/HAZMB-CAS  0.25
      MB/HAZMB+PEA -0.50
      MB/HAZXX+HAZ  1.0
      MD/CAS$$$$$$  0.25
      MD/CASXX+CAS  1.0
      MD/PEA$$$$$$  0.25
      MD/PEAXX+PEA  1.0
      MD/HAZ$$$$$$  0.25
      MD/HAZXX+HAZ  1.0
*
*     SLACK PRICES
      XX+CAS$$$$$$  0.65
      XX+PEA$$$$$$  0.25
      XX+HAZ$$$$$$  0.35
ENDATA
LISTPR      ; OUTPUT LABELS, COSTS, AND STATUS FLAGS
MATRIX      ; OUTPUT MATRIX TABLEAU WITHOUT UNIT POSITIVES.
EQLIST      ; OUTPUT MATRIX IN EQUATION FORM.
MIXMAP      ; OUTPUT MATRIX IN COMPACT, CODED FORM.
GOGOGO      ; OBTAIN OPTIMAL, FEASIBLE;RATIONAL SOLUTION.
DPMODE      ; INVERT TO CURRENT BASIS IN DOUBLE PRECISION.
SPMODE      ; RETURN TO SINGLE PRECISION.
GOGOGO      ; RESOLVE IN CASE INVERSION INCOMPLETE.
PRIMAL      ; OUTPUT ACTIVITY LEVELS AT SOLUTION.
DUAL        ; OUTPUT SHADOW PRICES.
REDCST      ; OUTPUT REDUCED COSTS OF NON BASIS VARIABLES.
RANGES      ; OUTPUT PRIMAL AND DUAL BASIS.
ERRORS      ; COMPUTE AND OUTPUT PRIMAL AND DUAL ERRORS.
ENDJOB      ; RETURN TO MONITOR.
∇ FIN
```

SAMPLE OUTPUT

```
XQT 1107LP                                                       .000.000.001


SET CASE1 TO (NUT-MI)     DEFINE CONTENTS OF CASE1
SET CASE2 TO (X     )     DEFINE CONTENTS OF CASE2
TITLE                     NEXT CARD CONTAINS PAGE HEADING
LOAD                      LOAD SHARE FORMAT MATRIX


 1107LP RUN ON NUT-MIX PROBLEM                                   .000.001.001

LOADER STATISTICS
STRUCTURAL MATRIX
   7 ROWS     9 COLUMNS     21 NON-ZERO ELEMENTS  33.3 PCT. DENSE
      .250000 IS SMALLEST ELEMENT       1.000000 IS LARGEST ELEMENT

COST VECTOR
   12 NON-ZERO ELEMENTS  75.0 PCT. DENSE
      .250000 IS SMALLEST ELEMENT        .650000 IS LARGEST ELEMENT

RHS VECTOR
   3 NON-ZERO ELEMENTS  42.9 PCT. DENSE
     .600000 IS SMALLEST ELEMENT        1.000000 IS LARGEST ELEMENT
   0 INITIAL INFEASIBILITIES    0 INITIAL ARTIFICIALS


 1107LP RUN ON NUT-MIX PROBLEM                                   .000.003.001


LIST PRINT REPORT


FLAG FUNCTION
   1-ORIGINAL BASIS
   2-CURRENT BASIS
   3-DESIRED BASIS
   4-FREE VARIABLE
   5-FROZEN VARIABLE
   6-INVERSION CONTROL
```

continued

```
PVNO IS THE PIVOT VECTOR COUNT AT THE TIME THE VARIABLE LEFT THE BASIS


    LABEL      COST       FLAG   1   2   3   4   5   6     PVNO
  1 MA-CAS      .000000          YES YES NO  NO  NO  NO       0
  2 MA+PEA      .000000          YES YES NO  NO  NO  NO       0
  3 MB-CAS      .000000          YES YES NO  NO  NO  NO       0
  4 MB+PEA      .000000          YES YES NO  NO  NO  NO       0
  5 XX+CAS      .650000          YES YES NO  NO  NO  NO       0
  6 XX+PEA      .250000          YES YES NO  NO  NO  NO       0
  7 XX+HAZ      .350000          YES YES NO  NO  NO  NO       0
  8 MA/CAS      .500000          NO  NO  NO  NO  NO  NO       0
  9 MA/PEA      .500000          NO  NO  NO  NO  NO  NO       0
 10 MA/HAZ      .500000          NO  NO  NO  NO  NO  NO       0
 11 MB/CAS      .350000          NO  NO  NO  NO  NO  NO       0
 12 MB/PEA      .350000          NO  NO  NO  NO  NO  NO       0
 13 MB/HAZ      .350000          NO  NO  NO  NO  NO  NO       0
 14 MD/CAS      .250000          NO  NO  NO  NO  NO  NO       0
 15 MD/PEA      .250000          NO  NO  NO  NO  NO  NO       0
 16 MD/HAZ      .250000          NO  NO  NO  NO  NO  NO       0

END OF LIST PRINT.
```

```
 1107LP RUN ON NUT-MIX PROBLEM                                  .000.005.001


    CASE NUT-MIX                      MATRIX TABLEAU

         LABEL       COST         BASIS         MA/CAS        MA/PEA        MA/HAZ
         COST      .000000       .000000       .500000       .500000       .500000
      ROW LABEL

       1 MA-CAS    .000000       .000000      -.500000       .500000       .500000
       2 MA+PEA    .000000       .000000      -.250000       .750000      -.250000
       3 MB-CAS    .000000       .000000       .000000       .000000       .000000
       4 MB+PEA    .000000       .000000       .000000       .000000       .000000
       5 XX+CAS    .650000      1.000000      1.000000       .000000       .000000
       6 XX+PEA    .250000      1.000000       .000000      1.000000       .000000
       7 XX+HAZ    .350000       .600000       .000000       .000000      1.000000
```

continued

```
1107LP RUN ON NUT-MIX PROBLEM                                     .000.006.001

   CASE NUT-MIX                   MATRIX TABLEAU

         LABEL      MB/CAS       MB/PEA       MB/HAZ       MD/CAS       MD/PEA
         COST      .350000      .350000      .350000      .250000      .250000
      ROW LABEL

        1 MA-CAS   .000000      .000000      .000000      .000000      .000000
        2 MA+PEA   .000000      .000000      .000000      .000000      .000000
        3 MB-CAS  -.750000      .250000      .250000      .000000      .000000
        4 MB+PEA  -.500000      .500000     -.500000      .000000      .000000
        5 XX+CAS  1.000000      .000000      .000000     1.000000      .000000
        6 XX+PEA   .000000     1.000000      .000000      .000000     1.000000
        7 XX+HAZ   .000000      .000000     1.000000      .000000      .000000

1107LP RUN ON NUT-MIX PROBLEM                                     .000.007.001

   CASE NUT-MIX                 MATRIX TABLEAU

         LABEL      MD/HAZ
         COST      .250000
      ROW LABEL

        1 MA-CAS   .000000
        2 MA+PEA   .000000
        3 MB-CAS   .000000
        4 MB+PEA   .000000
        5 XX+CAS   .000000
        6 XX+PEA   .000000
        7 XX+HAZ  1.000000

  END OF MATRIX TABLEAU
```

continued

```
 1107LP RUN ON NUT-MIX PROBLEM                                  .000.008.001

EQLIST         OUTPUT MATRIX IN EQUATION FORM.

 1107LP RUN ON NUT-MIX PROBLEM                                  .000.009.001

   CASE NUT-MIX                     EQLIST OUTPUT

   EQUATION     1 LABEL MA-CAS COST =      .000000 ORIGINAL BI =        .000000

   +   1.00000(MA-CAS) =   .50000(MA/CAS) +   .50000(MA/PEA) +   .50000(MA/HAZ)

   EQUATION     2 LABEL MA+PEA COST =  .000000 ORIGINAL BI =   .000000

   +   1.00000(MA+PEA) =    .25000(MA/CAS) +  .75000(MA/PEA) =   .25000(MA/HAZ)

   EQUATION     3 LABEL MB-CAS COST =      .000000 ORIGINAL BI =        .000000

   +   1.00000(MB-CAS) =    .75000(MB/CAS) +   .25000(MB/PEA) +  .25000(MB/HAZ)

   EQUATION     4 LABEL MB+PEA COST =       .000000 ORIGINAL BI =       .000000

   +   1.00000(MB+PEA) =   .50000(MB/CAS) +    .50000(MB/PEA) + .50000(MB/HAZ)

   EQUATION     5 LABEL XX+CAS COST =       .650000 ORIGINAL BI =     1.000000

   +   1.00000(XX+CAS) +  1.00000(MA/CAS) +   1.00000(MB/CAS) + 1.00000(MD/CAS)

   EQUATION     6 LABEL XX+PEA COST =       .250000 ORIGINAL BI =  1.000000

   +   1.00000(XX+PEA) +  1.00000(MA/PEA) +   1.00000(MB/PEA) + 1.00000(MD/PEA)

   EQUATION     7 LABEL XX+HAZ COST =     .350000 ORIGINAL BI =     .600000

   +   1.00000(XX+HAZ) +  1.00000(MA/HAZ) +  1.00000(MB/HAZ) +  1.00000(MD/HAZ)

   END EQLIST OUTPUT
```

continued

```
1107LP RUN ON NUT-MIX PROBLEM                                  .000.011.001

                              MTXMAP OUTPUT

   CASE NUT-MIX      ITERATION    0 OBJECTIVE VALUE           .000000


              C B M M M M M M M M M
              O A A A A B B B D D D
              S S / / / / / / / / /
              T I C P H C P H C P H
                S A E A A E A A E A
                  S A Z S A Z S A Z

        COST  . . 5 5 5 5 5 5 5 5 5
      MA-CAS  . .-5 5 5 . . . . . .
      MA+PEA  . .-5 5-5 . . . . . .
      MB-CAS  . . . . .-5 5 5 . . .
      MB+PEA  . . . . .-5 5-5 . . .
      XX+CAS  5 1 1 . . 1 . . 1 . .
      XX+PEA  5 1 . 1 . . 1 . . 1 .
      XX+HAZ  5 5 . . 1 . . 1 . . 1

1107LP RUN ON NUT-MIX PROBLEM                                  .000.012.001

   THE MATRIX ELEMENTS ARE REPRESENTED AS FOLLOWS

    CODE  MAGNITUDE OF ELEMENTS  OCCURENCES

           GREATER  EQUAL TO OR
            THAN     LESS THAN


      0      0.0        0.0001         0
      2      0.0001     0.001          0
      3      0.001      0.01           0
      4      0.01       0.1            0
      5      0.1        0.9999        25
      1           =1.0                11
      6      1.0       10.0            0
      7     10.0      100.0            0
      8    100.0     1000.0            0
      9   1000.0                       0

    END MTXMAP OUTPUT
```

continued

1107LP RUN ON NUT-MIX PROBLEM .000.013.001

```
GOGOGO        OBTAIN OPTIMAL, FEASIBLE, RATIONAL SOLUTION.

STAGE    0  PVCNT  0  OBFVAL      1.110000 NET CHANGES    0
           96  MS FOR DOZ
ITCNT    0 DJBAD 1000.000 KOLAY  9 NOLAY    9 ART    0 INF    0
          135  MS FOR SSS
           26  MS FOR UPDATE
           65  MS FOR OPTMIZ
           78  MS FOR PIVGEN

STAGE    1  PVCNT  5  OBFVAL      1.160000 NET CHANGES    5
          220  MS FOR DOZ
ITCNT    5 DJBAD 1000.000 KOLAY  0 NOLAY    0 ART    0 INF    0
          111  MS FOR SSS
           49  MS FOR UPDATE
           32  MS FOR OPTMIZ
            4  MS FOR PIVGEN

STAGE    2  PVCNT  5  OBFVAL      1.160000 NET CHANGES    5

DPMODE        INVERT TO CURRENT BASIS IN DOUBLE PRECISION.
INVERSION FINISHED WITH    0 NOT REMOVED AND    0 NOT ENTERED
SPMODE        RETURN TO SINGLE PRECISION
GOGOGO        RESOLVE IN CASE INVERSION INCOMPLETE.

STAGE    0  PVCNT  5  OBFVAL      1.160000 NET CHANGES    5
           99  MS FOR DOZ
ITCNT    5 DJBAD 1000.000 KOLAY  9 NOLAY    9 ART    0 INF    0
          137  MS FOR SSS
           94  MS FOR UPDATE
           33  MS FOR OPTMIZ
            2  MS FOR PIVGEN

STAGE    1  PVCNT  5  OBFVAL      1.160000 NET CHANGES    5

PRIMAL        OUTPUT ACTIVITY LEVELS AT SOLUTION.
```

continued

```
1107LP RUN ON NUT-MIX PROBLEM                                    .000.014.001

                             PRIMAL OUTPUT

   CASE NUT-MIX      ITERATION    5 OBJECTIVE VALUE     1.160000


      LABEL        COST        ACTIVITY          LABEL        COST      ACTIVITY
      XX+PEA      .250000       .500000          XX+HAZ      .350000     .100000
      MA/CAS      .500000      1.000000          MA/PEA      .500000     .500000
      MA/HAZ      .500000       .500000          MB/PEA      .350000     .000000
      MB/HAZ      .350000       .000000

   END PRIMAL OUTPUT
```

```
1107LP RUN ON NUT-MIX PROBLEM                                    .000.016.001

                               DUAL OUTPUT

   CASE NUT-MIX      ITERATION    5 OBJECTIVE VALUE     1.160000


      LABEL        COST     SHADOW PRICE         LABEL       COST   SHADOW PRICE
      MA-CAS      .000000       .350000          MA+PEA      .00000     .100000
      MB-CAS      .000000       .200000          MB+PEA      .00000     .100000
      XX+CAS      .650000       .700000          XX+PEA      .25000     .250000
      XX+HAZ      .350000       .350000

   END DUAL OUTPUT
```

```
1107LP RUN ON NUT-MIX PROBLEM                                    .000.018.001

                              REDCST OUTPUT

   CASE NUT-MIX      ITERATION    5 OBJECTIVE VALUE     1.160000


      LABEL        COST     REDUCED COST         LABEL       COST   REDUCED COST
      MA-CAS      .000000       .350000          MA+PEA      .00000     .100000
      MB-CAS      .000000       .200000          MB+PEA      .00000     .100000
      XX+CAS      .650000       .050000          MB/CAS      .35000     .150000
      MD/CAS      .250000       .450000          MD/PEA      .25000     .000000
      MD/HAZ      .250000       .100000

    END REDCST OUTPUT
```

continued

```
1107LP RUN ON NUT-MIX PROBLEM                                  .000.020.001

                           DUAL RANGE OUTPUT

   CASE NUT-MIX      ITERATION    5 OBJECTIVE VALUE       1.160000


                           - - - - - - LIMITS OF RANGE - - - - - -
    LABEL   ORIG. ACT.    LABEL    INCREMENT   LABEL     INCREMENT
   MA-CAS      .000000 XX+HAZ     -.066667 MA/HAZ        .333333
   MA+PEA      .000000 MA/HAZ     -.500000 XX+HAZ        .100000
   MB-CAS      .000000 XX+HAZ     -.050000 MB/HAZ        .000000
   MB+PEA      .000000 MB/HAZ     -.000000 MB/PEA        .000000
   XX+CAS     1.000000 XX+HAZ     -.200000 MA/HAZ       1.000000
   MB/CAS      .000000 MB/HAZ     -.000000 XX+HAZ        .200000
   MD/CAS      .000000 XX+HAZ     -.200000 MA/HAZ       1.000000
   MD/PEA      .000000         -9999.000000 XX+PEA       .500000
   MD/HAZ      .000000         -9999.000000 XX+HAZ       .100000

   END DUAL RANGE OUTPUT

1107LP RUN ON NUT-MIX PROBLEM                                  .000.021.001

                          PRIMAL RANGE OUTPUT

   CASE NUT-MIX      ITERATION    5 OBJECTIVE VALUE       1.160000


                           - - - - - - LIMITS OF RANGE - - - - - -
    LABEL      COST       LABEL    INCREMENT   LABEL     INCREMENT
   XX+PEA      .250000 MD/PEA     -.000000 MA+PEA        .100000
   XX+HAZ      .350000 MA+PEA     -.100000 MB-CAS        .100000
   MA/CAS      .500000 XX+CAS     -.050000           9999.000000
   MA/PEA      .500000 MA+PEA     -.100000           9999.000000
   MA/HAZ      .500000 XX+CAS     -.100000 MA+PEA        .100000
   MB/PEA      .350000 MB-CAS     -.100000 MB/CAS        .075000
   MB/HAZ      .350000 MB-CAS     -.100000 MB+PEA        .100000

   END PRIMAL RANGE OUTPUT
```

continued

```
 1107LP RUN ON NUT-MIX PROBLEM                                  .000.023.001

                     DUAL ERROR ANALYSIS

      LABEL        Z(J)          C(J)             ERROR
      XX+PEA        .250          .250        .00000000
      XX+HAZ        .350          .350        .00000000
      MA/CAS        .500          .500        .00000000
      MA/HAZ        .500          .500        .00000000
      MB/PEA        .350          .350        .00000000
      MB/HAZ        .350          .350        .00000000

MAXIMUM DUAL ERROR OF    .00000000   FOR VARIABLE

 1107LP RUN ON NUT-MIX PROBLEM                                  .000.024.001

                   PRIMAL ERROR ANALYSIS

      LABLE   ORIG B(I)    CALC B(I)          ERROR
      MA-CAS        .000         .000    -.00000000
      MA+PEA        .000         .000    -.00000000
      MB-CAS        .000         .000    -.00000000
      MB+PEA        .000         .000    -.00000000
      XX+CAS       1.000        1.000    -.00000000
      XX+PEA       1.000        1.000    -.00000000
      XX+HAZ        .600         .600    -.00000000

MAXIMUM PRIMAL ERROR OF    .00000000    FOR VARIABLE

 1107LP RUN ON NUT-MIX PROBLEM                                  .000.025.001

ENDJOB        RETURN TO MONITOR.

RUN TIME  0 HRS  0 MIN 21 SEC
```

COST ESTIMATE

For the program listed on the Sample Input, computer time was 21 seconds. At the current rate for the University of Notre Dame ($480./hr. plus input-output charges), the chargeable computer cost was $4.18.

Charge to user = computer costs + postage and handling + network overhead
= $4.18 + $10.00 + network overhead
= $14.18 + network overhead

CONTENTS—LP1107

pages

1	Identification & Abstract
3	User Instructions
5–15	I/O
17	Cost—Contents

DESCRIPTIVE TITLE	UNIVAC 1107 PERT/COST, PERT/TIME System
CALLING NAME	PERT (library tape)
INSTALLATION NAME	The University of Notre Dame Computing Center
AUTHOR(S) AND AFFILIATION(S)	UNIVAC Division of Sperry Rand Corporation
LANGUAGE	SLEUTH II and COBOL
COMPUTER	UNIVAC 1107
PROGRAM AVAILABILITY	Listings available
CONTACT	Elizabeth Hutcheson, EIN Technical Representative, Computing Center, University of Notre Dame, Notre Dame, Ind. 46556 Tel. (219) 283-7784

FUNCTIONAL ABSTRACT

The UNIVAC 1107 PERT System provides for integrated time and cost planning and control of research and development programs through implementation of the "work package" costing concept of the Department of Defense/National Aeronautics and Space Administration. The system is composed of two major program modules, PERT/COST and PERT/TIME, which generate reports for the project being analyzed. These reports include activity and event reports, a Work package/Activity listing, a Charge number analysis, etc., from PERT/TIME, and reports on the structure of labor and material costs from PERT/COST.

REFERENCES

PERT/COST General Reference Manual, (UP-4013), (UNIVAC Division of Sperry Rand Corp., N.Y.).

UNIVAC 1107/1108 PERT and CPM: Techniques in the Project Management, (UP 3952), (UNIVAC Division of Sperry Rand Corp., N.Y.).

Available through the local UNIVAC representative or through the EIN Office at the cost of reproduction and mailing.

USER INSTRUCTIONS

Potential users are directed to the references listed below for a full discussion of the options and constraints in effect.

REFERENCES

PERT/COST General Reference Manual, (UP-4013), (UNIVAC Division of Sperry Rand Corp., N.Y.).

UNIVAC 1107/1108 PERT and CPM: Techniques in the Project Management, (UP 3952), (UNIVAC Division of Sperry Rand Corp., N.Y.).

Available through the local UNIVAC representative or through the EIN Office at the cost of reproduction and mailing.

SAMPLE INPUT and SAMPLE OUTPUT

Sample I/O is included in the references listed in the USER INSTRUCTIONS; interested persons are directed to these documents.

COST ESTIMATE

See references listed in USER INSTRUCTIONS.

CONTENTS—PERT

pages

1 Identification & Abstract
3 User Instructions
5 I/O
7 Cost—Contents

DESCRIPTIVE TITLE	Inquirer II System for Content Analysis
CALLING NAME	I/II
INSTALLATION NAME	Washington University Computing Facilities
AUTHOR(S) AND AFFILIATION(S)	J. Philip Miller Washington University Computing Facilities
LANGUAGE	PL/I
COMPUTER	S/360 under OS
PROGRAM AVAILABILITY	Source listings and complete documentation available from Documentation Librarian, Computing Facilities, Box 1152, Washington University, St. Louis, Mo. 63130
CONTACT	J. Philip Miller, Computing Facilities, Box 1152, Washington University, St. Louis, Mo. 63130 Tel.: (314) 863-0100 ext. 4041

FUNCTIONAL ABSTRACT

The Inquirer II is a set of computer programs comparable to, but more flexible than, its predecessor, the General Inquirer, developed by Stone and his colleagues at Harvard. The original version of the General Inquirer System was designed for problems encountered in the content analysis of textual and verbal data. The General Inquirer was implemented for the IBM 7090-7094 computer along with the IBM 1401 computer. A later version of the General Inquirer was designed by Psathas and Miller to be used only with an IBM 1401 computer with an IBM 1311 disk drive. Stone originally described the General Inquirer as "a set of computer programs to (a) identify, systematically within text, instances of words and phrases that belong to categories specified by the investigator; (b) count occurrences and specify co-occurrences of these categories; (c) print and graph tabulations; (d) perform statistical tests; (e) sort and regroup sentences according to whether they contain instances of a particular category of combination of categories." The Inquirer II contains these capabilities and also

continued

allows for more elaborate analysis of the data. The I/II is able to make more elaborate contextual searches of the data and provide more options to the potential users. These options are described in detail in the *Inquirer II Programmer's Guide*.

Content analysis may be defined as a research technique which includes a systematic identification of theoretically relevant constructs in textual data. Content analysis is usually performed so that inferences can be made about the source or originator of the message, the message itself or the intended receiver of the message. The investigator communicates the constructs and the rules by which they may be identified within the corpus of text by means of dictionary. This dictionary is either one of his own construction or one which has been utilized in previous research.

REFERENCES

Miller, J. Philip, editor, *Inquirer II Programmer's Guide*, 2nd Edition, 1970, Washington University, (Available from Documentation Librarian, Computing Facilities, Box 1152, Washington University, St. Louis, Mo. for $5.00)

Stone, P.J., Dunphy, D.C., Smith, M.S., and Ogilvie, D.M. The General Inquirer: *A Computer Approach to Content Analysis*, Cambridge, Mass., The MIT Press, 1966.

USER INSTRUCTIONS

The primary inputs to the system are the data to be analyzed and a dictionary of concepts.

The major task in using the Inquirer II system is the creation of a dictionary of content analysis categories, called *concepts*. A concept consists of a number of language signs that together represent a variable in the investigator's theory.

The Sample Output appended gives a general idea of the process of presenting the data and formulating the dictionary of concepts. A thorough understanding for the user will be obtained by sending for the complete documentation, in particular the *Inquirer II Programmer's Guide*.[1]

REFERENCES

1. Miller, J. Philip, editor, *Inquirer II Programmer's Guide*, 2nd Edition, 1970, Washington University, (Available from Documentation Librarian, Computing Facilities, Box 1152, Washington University, St. Louis, Mo. for $5.00).

SAMPLE INPUT

```
QQA004I *    NEW TITLE ENCOUNTERED


 $Renee's delusional system from Autobiography of a Schizophrenic Girl, in Landis p 182-3$

 <<F15>>

 <p1> Very soon after the beginning of analysis I understood that my fear was a cover for guilt, a quilt infinite and

awful. During the early sessions, masturbation and the hostility I harbored toward everyone seemed to lie at the bottom.

I literally hated people, without knowing why. In dreams and frequently in waking fantasies I constructed an electric

machine to blow up the earth and everyone with it. This waq my greatest, most terrible revenge.


$Renee's delusional system from Autobiography of a Schizophrenic Girl, in Landis p 182-3$
<p1>
<<F15>>

QQA002I *    BEGINNING OF NEW DOCUMENT OF LEVEL  1



 <p2> Later, considering them appropriate, I no longer felt guilty about these fantasies, nor did the quilt have an

actual object . It was too pervasive, too enormous, to be founded on anything definite, and it demanded punishment. The

punishment was indeed horrible, sadistic --- it consisted, fittingly enough, of being guilty. For to feel oneself guilty

is the worst that can happen, it is the punishment of punishments. Consequently, I could never be relieved of it as

though I had been truly punished. Quite the reverse, I felt more and more guilty, immeasurably guilty. Consequently, I

sought to discover what was punishing me so dreadfully, what was making me so guilty.
```

SAMPLE OUTPUT

```
                              KWIC of data file

other beings. A formidable interdependence bound all men under the scourge of culpability. Everyone  F15*
f it as some vast world-like entity encompassing all men. At the top were those who gave orders, who F15*
were guilty. Since every man was responsible for all other men, each of his acts had a repercussion  F15*
***********       3 INSTANCES OF All


s desk like a puppet, separated from everything, alone under a blinding light, waving his arms like  F15*
***********       1 INSTANCES OF Alone


never deliver me of the load. Because, as I have already said, the most dreadful punishment was to m F15*
***********       1 INSTANCES OF Already


ty about these fantasies, nor did the guilt have an actual object . It was too pervasive, too enormo F15*
and frequently in waking fantasies I constructed an electric machine to blow up the earth and everyo F15*
 of my words seemed strange. Every now and then, an innner voice interrupted sneeringly, "Ah, Ah!"an F15*
***********       3 INSTANCES OF An


                Very soon after the beginning of analysis I understood that my fear was a cover for  F15*
***********       1 INSTANCES OF Analysis


         It was only when I was near "Mama," my analyst, that I felt a little better. ,,,I saw the  F15*
***********       1 INSTANCES OF Analyst


 my fear was a cover for guilt, a guilt infinite and awful. During the early sessions, masturbation  F15*
tructed an electric machine to blow up the earth and everyone with it. This was my greatest, most te F15*
lly hated people, without knowing why. In dreams and frequently in waking fantasies I constructed an F15*
oo enormous, to be founded on anything definite, and it demanded punishment. The punishment was inde F15*
an innner voice interrupted sneeringly, "Ah, Ah!"and mockingly repeated what I had said.             F15*
n truly punished. Quite the reverse, I felt more and more guilty, immeasurably guilty. Consequently, F15*
ndependence of each part that inspired such fear and prevented my recognizing her even though I knew F15*
o relate what had happened sinc e the last visit and relived it in the telling. But the sound of my  F15*
d awful. During the early sessions, masturbation and the hostility I harbored toward everyone seemed F15*
ved it in the telling. But the sound of my voice and the meaning of my words seemed strange. Every n F15*
th, then the nose, then the cheeks, then one eye and the other. Perhaps it was this independence of  F15*
 everything was there, posed, congealed, stupid. And the terror, the mad anguish, mounted in me.     F15*
diculous. "Ah, ah! then the teacher said, said," and the voice dwelt stiltedly on "said, said."I str F15*
he meaning of my words seemed strange. Every now and then, an innner voice interrupted sneeringly, " F15*
y, abominable, intolerably guilty, without cause and without motive. Any punishment, the very worst, F15*
a blinding light, waving his arms like a maniac. And I saw my little sister, rolling on the kitchen  F15*
***********      16 INSTANCES OF And


osed, congealed, stupid. And the terror, the mad anguish, mounted in me.                             F15*
***********       1 INSTANCES OF Anguish


erably guilty, without cause and without motive. Any punishment, the very worst, could be imposed on F15*
***********       1 INSTANCES OF Any
```

continued

Dictionary Compiler output--
Concept Name Paragraph

```
BEGIN COMPILATION OF SOURCE PROGRAM
                             *      AGE_TIME=01.
                             *      SOM_PSYCH_REF=02.
                             *      BODY_PARTS=02,03.
                             *      INT_BDY_PTS=02,03,04.
                             *      EXT_BDY_PTS=02,03,05.
                             *      GEN_BDY_PTS=02,03,06.
                             *      SOM_COND=02,07.
                             *      HEALTH=02,07,08.
                             *      ILLNESS=02,07,09.
                             *      DEATH=02,07,10.
                             *      TREATMENTS=11.
                             *      TESTS=12.
                             *      NON_SPEC_REF=13.
                             *      OBJ_ORIENT=14.
                             *      RETAIN=14,15.
                             *      EXPELL=14,16.
                             *      ATTAIN=14,17.
                             *      GENDER=18.
                             *      MALE=18,19.
                             *      FEMALE=18,20.
                             *      NEUTER=18,21.
                             *      AUTH_FIG=22.
                             *      TREAT_FIG=22,23.
                             *      LEGAL_FIG=22,24.
                             *      SENSE=25.
                             *      VISUAL=25,26.
                             *      AUDITORY=25,27.
                             *      GUSTATORY=25,28.
                             *      OLFACTORY=25,29.
                             *      DERMAL=25,30.
                             *      ACTION_NORM=31.
                             *      PASSIVE=31,32.
                             *      ACTIVE=31,33.
                             *      SEXUAL_ACT=34.
                             *      SEX_ACT=34,35.
                             *      FORE_PLAY=34,36.
                             *      PRE_FORE_PLAY=34,37.
                             *      SOCIAL_ACT=38.
                             *      AGGRESSIVE=38,39.
                             *      FRIENDLY=38,40.
                             *      ISOLATE=38,41.
                             *      COMMUNICATE=38,42.
                             *      DOMINATE=38,43.
                             *      SUBMIT=38,44.
                             *      HELP=38,45.
                             *      DIRECTION=46.
                             *      APPROACH=46,47.
                             *      AVOID=46,48.
                             *      ACHIEVE=49.
                             *      SUCCESS=49,50.
                             *      FAIL=49,51.
                             *      COG_PROC=52.
                             *      CONTEMPLATE=52,53.
                             *      COG_AWARE=52,54.
                             *      UNCERTAIN=52,55.
                             *      DECIDE=52,56.
                             *      REGARD=57.
                             *      POSITIVE=57,58.
                             *      NEGATIVE=57,59.
                             *      FORTUNE=60.
                             *      GOOD_FORT=60,61.
                             *      BAD FORT=60,62.
```

continued

Dictionary Compiler-- List of Concept
Name Paragraph and start of Entry Name Paragraph

```
*      ANGER=63,64.
*      LOVE=63,65.
*      HATE=63,66.
*      FEAR=63,67.
*      HAPPY=63,68.
*      SAD=63,69.
*      DISTRESS=63,70.
*      EMOTION_POS=63,71.
*      EMOTION_NEG=63,72.
*      UNDEFINED_NEED=73.
*      OBSTACLES=74.
*      OBST_PRES=74,75.
*      STRUGGLE=74,76.
*      SELF=77.
*      SPEC_OTHERS=78.
*      NON_SPEC_OTHERS=79.
*       SENT_LENGTH = 90.
*       LONG_SENT = 90,91.
*       SHORT_SENT = 90,92.
*       SENT_TYPE = 93.
*       QUESTION = 93,94.
*       DECLARE = 93,95.
*      ABOMINABLE  :  NEGATIVE.
*      ABOUT  :  IF WORD (-4,-2) = 'UP' & WORD (-3,-1) = 'AND'
*                THEN HEALTH
*                ELSE IF WORD (-2,-1) = 'BRING'|
*                        WORD (-2,-1) = 'BROUGHT'|
*                        WORD (-2,-1) = 'COME'
*                THEN SUCCESS
*                ELSE IF WORD (1,2) = 'TO'
*                THEN ACTIVE
*                ELSE.
*      ACT:     IF WORD (-2,-1) = 'SEXUAL' |
*                   WORD (-2,-1) = 'SEX'
*                THEN SEX_ACT
*                ELSE IF WORD (1,1) = 'UP'
*                THEN AGGRESSIVE
*                ELSE IF WORD (1,1) = 'UPON'
*                THEN ACTIVE
*                ELSE ACTIVE.
*      ALL:      IF ORDER(WORD (-3,-2) = 'FREE', WORD (-2,-1) = 'FOR')
*                THEN AGGRESSIVE
*                ELSE IF WORD (1,3) = 'OVER'
*                THEN ACHIEVE
*                ELSE.
*      ALONE:    ISOLATE.
*      ANALYSIS:  CONTEMPLATE.
*      ANALYST:  TREAT_FIG; NEUTER.
*      ANGUISH:  SAD; EMOTION_NEG.
*      ANY:  NON_SPEC_REF.
*      ANYTHING:  NON_SPEC_REF.
*      APPARENT:  DECIDE.
*      APPARENTLY:
*      APPROPRIATE:  POSITIVE.
*      ARM:  EXT_BDY_PTS.
*      ARMS:     IF WORD (-3,-1) = 'OPEN'
*                THEN FRIENDLY
*                ELSE IF WORD (-3,-2) = 'UP' & WORD (-2,-1) = 'IN'
*                THEN AGGRESSIVE
*                ELSE EXT_BDY_PTS.
*      ASSOCIATE:  IF WORD (-2,-1) = 'AN' |
*                     WORD (-2,-1) = 'THE'
*                  THEN
*                  ELSE CONTEMPLATE.
*      AT:       IF WORD (-3,-1) = 'FLY' | WORD (-3,-1) = 'FLEW'
*                THEN AGGRESSIVE
```

continued

Portion of Entry Name Paragraph of Dictionary

```
*              ELSE IF WORD (1,2) = 'WELL'
*              THEN HEALTH
*              ELSE.
*     FELT:    IF WORD (1,5) = 'FOR'
*              THEN (LOVE; EMOTION_POS)
*              ELSE IF WORD (1,2) = 'WELL'
*              THEN HEALTH
*              ELSE.
*     FIGURE:  IF WORD (-2,-1) = 'A' | WORD (-2,-1) = 'THE' |
*                 WORD (-2,-1) = 'THIS' | WORD (-2,-1) = 'THESE' |
*                 WORD (-2,-1) = 'THOSE'
*              THEN
*              ELSE IF WORD (1,3) = 'OUT'
*              THEN DECIDE
*              ELSE CONTEMPLATE.
*     FIT:     IF WORD (-2,-1) = 'THROW' | WORD (-5,-1) = 'HAVE' |
*                 WORD (-5,-1) = 'HAD'
*              THEN AGGRESSIVE
*              ELSE IF WORD (1,2) = 'IN' & WORD (2,3) = 'WITH'
*              THEN SUBMIT
*              ELSE.
*     FOUND:   IF WORD (1,2) = 'OUT'
*              THEN SUCCESS
*              ELSE GOOD_FORT.
*     FROM:    IF WORD (-3,-1) = 'SHRANK' | WORD (-3,-1) = 'SHRINK' |
*                 WORD (-3,-1) = 'SHRUNK' | ORDER(WORD (-3,-1) = 'GET',
*                 WORD (-2,-1) = 'AWAY') | WORD (-3,-1) = 'REACH' |
*                 WORD (-3,-1) = 'RAN' | WORD (-3,-1) = 'RUN'
*              THEN AVOID
*              ELSE IF WORD (-2,-2) = 'GET' & WORD (-1,-1) = 'BACK'
*              THEN APPROACH
*              ELSE.
*     GAVE:    IF WORD (1,2) = 'IN' | WORD (1,3) = 'UP' | WORD (1,5) = 'WAY'
*              THEN SUBMIT
*              ELSE EXPELL.
*     GET:     IF WORD (1,3) = 'AROUND' | WORD (1,2) = 'UP'
*              THEN ACTIVE
*              ELSE IF WORD (1,3) = 'AWAY' | WORD (1,3) = 'OUT' |
*                      WORD (1,3) = 'OFF'
*              THEN AVOID
*              ELSE IF WORD (1,2) = 'BACK' | WORD (1,2) = 'IN' |
*                      WORD (1,2) = 'ON' | WORD (1,1) = 'TO' |
*                      WORD (1,1) = 'AROUND' & WORD (2,2) = 'TO' |
*                      WORD (1,1) = 'DOWN' & WORD (2,2) = 'TO'
*              THEN APPROACH
*              ELSE IF ORDER(WORD (1,2) = 'HANG', WORD (2,3) ='OF') |
*              ORDER(WORD (1,3) = 'TO', WORD (2,4) = 'BOTTOM',
*              WORD (3,5) = 'OF') | WORD (1,2) = 'THROUGH' |
*              WORD (1,1) = 'AWAY' & WORD (2,2) = 'WITH'
*              THEN SUCCESS
*              ELSE IF WORD (1,1)= 'HOLD' & WORD (2,2) = 'OF' |
*              ORDER(WORD (1,4) = 'IN', WORD (2,5) = 'WAY')
*              THEN AGGRESSIVE
*              ELSE IF ORDER(WORD (1,2) = 'WIND', WORD (2,3) = 'OF') |
*              ORDER(WORD (1,3) = 'THING', WORD (2,4) = 'ACROSS',
*              WORD (3,5) = 'TO'   )
*              THEN COMMUNICATE
*              ELSE IF WORD (1,1) = 'RID' & WORD (2,2) = 'OF'
*              THEN EXPELL
*              ELSE IF WORD (1,1) = 'THE' & WORD (2,2) = 'BETTER' &
*              WORD (3,3) = 'OF'
*              THEN DOMINATE
*              ELSE IF WORD (1,1) = 'IN' & WORD (2,2) = 'ON'
*              THEN FRIENDLY
*              ELSE IF WORD (1,1) = 'OVER'
*              THEN OBST_PRES
```

continued

CONCEPTS: Listing of Tagged File

```
        Renee's delusional system from Autobiography of a Schizophrenic Girl

ID FIELD OF LEVEL   2 = P15

ID FIELD OF LEVEL   1 = p1

SENTENCE     1;  22 WORDS
CURRENT ID p1

Very     soon          after    the     beginning   of    analysis       I        understood     that
         AGE_TIME                       AGE_TIME          COG_PROC       SELF                    NON_SPEC_REF
                                                          CONTEMPLATE

my       fear              was    a     cover    for   guilt,          a    guilt             infinite    and
SELF     EMOTION_NEG                                   EMOTION_NEG          EMOTION_NEG
         EMOTION                                       EMOTION              EMOTION
         FEAR                                          FEAR                 FEAR

awful        .
REGARD
NEGATIVE

SENTENCE     2;  18 WORDS
CURRENT ID p1

During    the     early    sessions.    masturbation     and    the     hostility      I       harbored
                                        PRE_FORE_PLAY                   EMOTION_NEG    SELF
                                        SEXUAL_ACT                      ANGER
                                                                        EMOTION

toward      everyone           seemed    to    lie          at    the    bottom    .
DIRECTION   NON_SPEC_OTHERS                    SOCIAL_ACT
APPROACH    GENDER                             AGGRESSIVE
            NEUTER                             COMMUNICATE

SENTENCE     3;   7 WORDS
CURRENT ID p1

I       literally    hated          people,            without    knowing      why    .
SELF                 EMOTION_NEG    NON_SPEC_OTHERS               COG_AWARE
                     ANGER          GENDER                        DECIDE
                     EMOTION        NEUTER                        COG_PROC

SENTENCE     4;  21 WORDS
CURRENT ID p1

In     dreams             and    frequently    in    waking    fantasies    I       constructed    an
       UNDEFINED_NEED                                                      SELF    ACTIVE
                                                                                   ACTION_NORM

electric    machine    to    blow          up    the    earth    and    everyone           with
                             SOCIAL_ACT                                NON_SPEC_OTHERS
                             AGGRESSIVE                                GENDER
                                                                       NEUTER

it              .
NON_SPEC_REF

SENTENCE     5;   7 WORDS
CURRENT ID p1

This    was    my      greatest,    most    terrible    revenge    .
               SELF    REGARD               REGARD
                       POSITIVE             NEGATIVE
```

continued

Tabulation of tagged file

```
enee's delusional system from Autobiography of a Schizophrenic Girl, in Landis p 182-3

COPCEPT NAME               TOTAL   WORD      CONCEPT  WD - CONC      SENTENCE  SENTENCE  SNTC - CONC
                           FREQ    INDEX     INDEX    INDEX          FREQ      INDEX     INDEX
                                      75        57       29                        5         5
AUDITORY

GUSTATORY

OLFACTORY

DERMAL

PASSIVE

ACTIVE                        1    1.33      1.75     3.44              1      20.00     20.00

SEX_ACT

FORE_PLAY

PRE_FORE_PLAY                 1    1.33      1.75     3.44              1      20.00     20.00

AGGRESSIVE                    2    2.66      3.50     6.89              2      40.00     40.00

FRIENDLY

ISOLATE

COMMUNICATE                   1    1.33      1.75     3.44              1      20.00     20.00

DOMINATE

SUBMIT

HELP

APPROACH                      1    1.33      1.75     3.44              1      20.00     20.00

AVOID
```

continued

Tabulation of tagged file

```
Renee's delusional system from Autobiography of a Schizophrenic Girl, in Landis p 182-3

  CONCEPT NAME                 TOTAL   WORD      CONCEPT  WD - CONC      SENTENCE  SENTENCE  SNTC - CONC
                               FREQ    INDEX     INDEX    INDEX          FREQ      INDEX     INDEX
                                          75        57       29                        5         5

  SUCCESS

  FAIL

  CONTEMPLATE                      1    1.33      1.75     3.44              1      20.00     20.00

  COG_AWARE                        1    1.33      1.75     3.44              1      20.00     20.00

  UNCERTAIN

  DECIDE                           1    1.33      1.75     3.44              1      20.00     20.00

  POSITIVE                         1    1.33      1.75     3.44              1      20.00     20.00

  NEGATIVE                         2    2.66      3.50     6.89              2      40.00     40.00

  GOOD_FORT

  BAD_FORT

  ANGER                            2    2.66      3.50     6.89              2      40.00     40.00

  LOVE

  HATE

  FEAR                             3    4.00      5.26    10.34              1      20.00     20.00

  HAPPY

  SAD

  DISTRESS

  EMOTION_POS
```

COST ESTIMATE

It is difficult to offer a precise cost estimate for a given task. A full treatment of a 5000-word text would cost *approximately* $500.

CONTENTS—I/II

pages
1– 2 Identification & Abstract
3 User Instructions
5–12 I/O
13 Cost—Contents

DESCRIPTIVE TITLE	Basic Information Retrieval System
CALLING NAME	BIRS
INSTALLATION NAME	Michigan State University Information Systems Laboratory
AUTHOR(S) AND AFFILIATION(S)	John F. Vinsonhaler, Ph.D. John M. Hafterson Stuart W. Thomas, Jr. Michigan State University
LANGUAGE	USASI Full FORTRAN
COMPUTER	CDC 3600, CDC 6500, CDC 6600 IBM 360 G-level GE 600 Series, others
PROGRAM AVAILABILITY	Decks and listings are currently available at cost from Michigan State University for non-profit institutions. The program is distributed to profit-making institutions by Hygain Technologies, 65 Whitney Street, Westport, Conn. Maintenance for all users is provided by Hygain Technologies.
CONTACT	Dr. John F. Vinsonhaler, Director, Information Systems Laboratory, 309 Computer Center, Michigan State University, East Lansing, Mich. 48823 Tel.: (517) 353-7284

FUNCTIONAL ABSTRACT

BIRS is a general purpose system of programs for the behavioral sciences and education. Essentially, BIRS is a set of fundamental program modules designed to allow scholars and scientists to use their own locally based computer to construct and maintain a variety of information systems. Search, maintenance, and index creation are performed automatically. Thus, BIRS may be viewed as a set of essential tools; the research worker may use these tools to construct the type of information system which best meets his immediate needs.

continued

It is not economically practical for individual educators and social scientists to develop their own special purpose programming system; they must share the costs of system developments by exchanging programs. General purpose systems like BIRS (with machine, data, and application independent programming) are ideally suited for free exchange among computer users.

REFERENCES

Vinsonhaler, J.F., Ed., *Technical Manual for the Basic Indexing and Retrieval System, BIRS 2.0*, Michigan State University, East Lansing, Mich., 1968.

Vinsonhaler, J.F., and Hafterson, J.M., Eds., *Technical Manual for the Basic Indexing and Retrieval System, Appendix I, BIRS 2.5*, Michigan State University, East Lansing, Mich., 1969.

Thomas, Jr., S.W., and Vinsonhaler, J.F., Eds., *Technical Manual for the Basic Information Analysis System, BIRS Technical Manual, Appendix II, BIAS 1.0*, Michigan State University, East Lansing, Mich., 1969.

Documentation is available at cost from the Information Systems Laboratory and from Hygain Technologies.

USER INSTRUCTIONS

See the references cited below. The nature and complexity of the system can be studied in brief with the help of the samples of input and output following.

REFERENCES

Vinsonhaler, J.F., Ed., *Technical Manual for the Basic Indexing and Retrieval System, BIRS 2.0*, Michigan State University, East Lansing, Mich., 1968.

Vinsonhaler, J.F., and Hafterson, J.M., Eds., *Technical Manual for the Basic Indexing and Retrieval System, Appendix I, BIRS 2.5*, Michigan State University, East Lansing, Mich., 1969.

Thomas, Jr., S.W. and Vinsonhaler, J.F., Eds., *Technical Manual for the Basic Information Analysis System, BIRS Technical Manual, Appendix II, BIAS 1.0*, Michigan State University, East Lansing, Mich., 1969.

Thomas, Jr., S.W., Hafterson, J.M., and Vinsonhaler, J.F., *Some Observations on the Structure and Operation of an Information Analysis System: Paper of the Institute No. 62*, (Michigan State University, East Lansing, Mich., 1969).

Vinsonhaler, J.F., Hafterson, J.M., and Thomas, Jr., S.W., "BIRS, A General Purpose Information Management System for Education and Social Science," *Proceedings of SHARE XXXIV*, (1970).

Documentation is available at cost from the Information Systems Laboratory and from Hygain Technologies.

SAMPLE INPUT AND OUTPUT

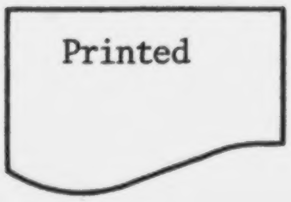
Information Storage: Creating the Information File

INPUT FOR BIRS PROCESSING

```
*$IFMP
*$DELIMITER IS ≠
*$NEW FILE
*$ABSTRACT
  ≠STATE   MICHIGAN
  ≠PUBLIC LAW   89-10
  ≠TITLE   AUGMENTATION OF SPECIAL EDUCATION AND PRE-EDVOCATIONAL TRAINING
  PROGRAMS.

  ≠DESCRIPTORS   EMOTIONALLY DISTURBED,   READING DISABILITY,
  PREVOCATIONAL EDUCATION,   RESIDENTIAL SCHOOL
  ≠TOTAL CHILDREN   90
  ≠TOTAL EXPENDITURE   $47,000
  ≠INSTRUCTION SALARIES = $39,658,   MATERIALS = $1,542,   OTHER = 0
  ≠PROJECT PARTICIPANTS
                            TMR      EMR      HH
  NUMBER OF TEACHERS         5        3        0
  OTHER PERSONNEL            0        0        2

  ≠SUMMARY
       THIS PROJECT IS TO REDUCE THE NUMBER OF GRADE LEVELS PER CLASSROOM
  IN OUR SCHOOL  PROVIDE GREATER INDIVIDUALIZATION OF INSTRUCTION WITHIN
  THE CLASSROOMS AND, TO EXPOSE OUR EMOTIONALLY DISTURBED CHILDREN TO
  GREATER SOCIALIZATION PROGRAMS
          .
          .
          .
      other data elements preceded by *$ABSTRACT
          .
          .
          .
*$END
```

OUTPUT FROM BIRS PROCESSING

INFORMATION FILE — CONTAINING DATA RECORDS



continued

Information Indexing: Preparing a Key Word Index

INPUT FOR BIRS PROCESSING

```
*$PIP

*$DELIMITER  ≠

*$USE FIELDS

     PUBLIC LAW,STATE
*$BOOK INDEX
*$END
```

OUTPUT FROM BIRS PROCESSING

Index for the Information File

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

(PUBLIC LAW)

     89-10           1, 22-60

     89-313          2-21, 100-350

     TITLE IV        50, 53, 500-650

(STATE)

     ALABAMA         200-250, 301

     ALASKA          10-11, 22
          .
          .
          .
          .
          .
```

continued

Information Retrieval: Preparing a List of Particular Types of Projects

---

INPUT FOR BIRS PROCESSING

```
*$DFSP
*$DELIMITER  ≠
*$QUESTION
  ≠STATE,MICHIGAN.AND.(#DESCRIPTORS,EMOTIONALLY DISTURBED.OR.
  EMOTIONALLY HANDICAPPED)
*$END
*$IFRP
*$DELIMITER  ≠
*$PRINT LINES 1 TO 5
*$END
```

OUTPUT FROM BIRS PROCESSING

SEARCH REPORT

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

  ABSTRACT 1      RELEVANCE 1.00
  ≠STATE   MICHIGAN
  ≠PUBLIC LAW   89-10
  ≠TITLE   AUGMENTATION OF SPECIAL EDUCATION AND PRE-VOCATIONAL TRAINING
  PROGRAMS


  ABSTRACT 35    RELEVANCE 1.00
  ≠STATE   MICHIGAN
  ≠PUBLIC LAW   TITLE IV-A
  ≠TITLE   IMPROVEMENT OF SPECIAL ASSISTANCE FOR EMOTIONALLY DISTURBED
```

continued

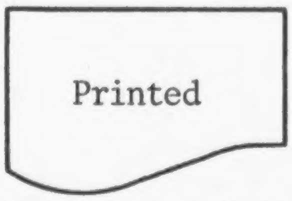
Information Analysis: Preparing a Statistical Report

INPUT FOR BIRS PROCESSING

```
*$IRPG
*$DELIMITER  ≠

*$DIMENSION
STATES = ≠STATE,MICHIGAN,HAWAII,ILLINOIS,MISSOURI

*$DIMENSION
PURPOSE = ≠INSTRUCTION,SALARIES,MATERIALS,TOTAL(SALARIES,MATERIALS)

*$DEFINE DATA
COST = .SUM.≠INSTRUCTION

*$TABLE
ALLOCATION OF FUNDS FOR INSTRUCTION (COST) = PURPOSE,STATES
```

OUTPUT FROM BIRS PROCESSING

TABULAR REPORT



```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

TABLE---ALLOCATION OF FUNDS FOR INSTRUCTION
DATA----COST
ROWS----SALARIES, MATERIALS, TOTAL
COLUMNS-MICHIGAN, HAWAII, ILLINOIS, MISSOURI

-------------------------------------------------------------------------------
SECTION   1   OF   1   SECTION(S)                                 PAGE 1
-------------------------------------------------------------------------------

              MICHIGAN      HAWAII      ILLINOIS      MISSOURI

SALARIES      80385.00     69717.00     71800.00      41154.00
MATERIALS      4642.00       826.00      5600.00       1011.00
TOTAL         85027.00     70543.00     77400.00      42165.00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

COST ESTIMATE

Unfortunately, cost estimates for the sample program are unavailable. The average cost for complete processing is about $15.00 per abstract per year for most applications, including file maintenance, searching, and report generation.

Charge to user = computer costs + network overhead

= $15.00/abstract per year + network overhead

CONTENTS—BIRS

page	
1–2	Identification & Abstract
3	User Instructions
5–8	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	Matching Factor Solutions
CALLING NAME	MATCHFS
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service (ETS)
AUTHOR(S) AND AFFILIATION(S)	Program due to: R. Pennell, ETS Adaptation due to: J. Barone, ETS
LANGUAGE	F4STAT (Statistically augmented FORTRAN IV)
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Office of Data Analysis Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

It frequently happens that two factor solutions are obtained in a study, and the question arises as to the extent of similarity. Cliff's[1] approach to the problem is succinctly stated in the abstract of his paper,

> Two problems are considered. The first is that of rotating two factor solutions orthogonally to a position where corresponding factors are as similar as possible. A least-squares solution for transformations of the two factor matrices is developed. The second problem is that of rotating a factor matrix orthogonally to a specified target matrix.

The present program performs least squares, procrustes rotations on a target (T) and a data (D) matrix following the general approach of Cliff.

General Description

The mathematical basis for the program is summarized briefly in the following outline.

continued

Case I: Find an A and B such that transformation $(E'E) = \text{minimum}$

where

$$E = TA - DB \ .$$

A and B are given by

$$T'D = Q$$
$$Q = A\Gamma B'$$

where A , B' and Γ are the Eckart and Young[2] decomposition of Q .

Case II: Find a C such that transformation $(E'E) = \text{minimum}$

where

$$E = T - DC$$

C is given by

$$C = B'A'$$

Case II is the typical target-matching situation were T is held fixed and D rotated to a least-squares fit.

The program can handle matrices as large as 125 rows (variables) X 50 columns.

REFERENCES

1. Cliff, N., "Orthogonal Rotation to Congruence," Psychometrika, *31*, pp. 33–42, (1966).

2. Eckart, C.T., and Young, G., "The Approximation of One Matrix by Another of Lower Rank," Psychometrika, *1*, pp. 211–218, (1936).

3. Pennell, R.J., and Young, F.W., "An IBM System/360 Program for Orthogonal Rotation to Congruence," Behavioral Science, *12*, p. 165, (1967).

USER INSTRUCTIONS

Card Preparation

Problem Card

Columns	*Parameter*	*Format*	*Description*
1– 5	NR	I5	Number of rows (variables) $\leq$ 125
6–10	NCT	I5	Number of columns in target matrix ($\leq$ 50)
11–15	NCD	I5	Number of columns in data matrix ($\leq$ 50)
16–20	ICASE	I5	Case option 1: Simultaneous rotations 2: Target matching 0: Both 1 and 2
21–25	IFMT	I5	Format option 0: Use same format for target and data matrices ≠ 0: Read in a different format for the data matrix

Title Card

Name of problem in Cols. 1–80

Format Card for Target Matrix

The format specification may start in any column. The word FORMAT is not used.

Target Matrix Cards

The program will read the punched data, row by row, according to the format specified for a row in Format Card for Target Matrix.

Format Card for Data Matrix

(Omit this card if IFMT = 0; the program will read the data matrix using the format specified in Format Card for Target Matrix.)

Data Matrix Cards

The program will read the punched data, row by row, according to the format specified for a row in Format Card for Data Matrix (Format Card for Target Matrix if IFMT = 0).

continued

System Control Cards

JOB Card will be prepared

System Card 1

Columns	*Contents*
1–37	`//STEPNAME EXEC GITNGO,NAME=MATCHFS`

System Card 2

Columns	*Contents*
1–19	`//GO.SYSIN DD *`

System Card 3

Columns	*Contents*
1– 2	`//`

Job Deck

JOB Card
System Card 1
System Card 2
Problem Card
Title Card
Format Card for Target Matrix
Target Matrix Cards
Format Card for Data Matrix (optional)
Data Matrix Cards
Blank Card
System Card 3

May be repeated for additional problems

SAMPLE INPUT

To illustrate the use of the program some simple artificial data were contrived. One factor solution describes four tests in terms of two factors, while the other solution shows the four tests in terms of three factors. Both Case I, for simultaneous rotations, and Case II, for target matching are illustrated.

```
//              JOB  (....
//STEPNAME   EXEC  GITNGO,NAME=MATCHFS
//GO.SYSIN   DD    *
     4    2    3    0    0
(3F8.3)
    0.700   0.100
    0.800   0.000
    0.100   0.700
    0.000   0.800
    0.760   0.320   0.500
    0.500   0.500  -0.400
    0.520  -0.360   0.500
    0.500  -0.500  -0.400

//
```

```
SAMPLE OUTPUT

CASE 1     SIMULTANEOUS ROTATIONS

TRANSFORMATION FOR FIRST MATRIX     AFTER ROTATION

            1          2                       1          2
   1      0.785     -0.619            1      0.611     -0.355
   2      0.619      0.785            2      0.628     -0.496
                                      3      0.512      0.488
                                      4      0.496      0.628

TRANSFORMATION FOR SECOND MATRIX    AFTER ROTATION

            1          2                       1          2
   1      0.994      0.059            1      0.818     -0.267
   2      0.060     -0.998            2      0.492     -0.476
   3      0.086      0.016            3      0.539      0.398
                                      4      0.433      0.522


CASE 2     FIRST MATRIX AS A TARGET

TRANSFORMATION TO TARGET            AFTER ROTATION

            1          2                       1          2
   1      0.744      0.662            1      0.808      0.297
   2      0.665     -0.747            2      0.682     -0.069
   3      0.058      0.066            3      0.177      0.646
                                      4      0.016      0.678

COEFFICIENTS OF CONGRUENCE
0.9861    0.9741
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed, and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $2.30.

Charge to user = computer costs + postage and handling + network overhead

= $2.30 + $5.00 + network overhead

= $7.30 + network overhead

CONTENTS—

pages	
1– 2	Identification & Abstract
3– 4	User Instructions
5– 6	I/O
7	Cost—Contents

000 0061

DESCRIPTIVE TITLE	Matrix Decomposition for Points of View Analysis
CALLING NAME	MATDEC
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service
AUTHOR(S) AND AFFILIATION(S)	J. Ferris Educational Testing Service
LANGUAGE	FORTRAN IV
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Deck and listings presently available
CONTACT	Mr. Ernest Anastasio, Off. of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

The purpose of MATDEC is to decompose a rectangular matrix (i.e., a data matrix, X, dimensioned number of variables by number of subjects) into three matrices U, Γ, and W, according to Horst's[1] development (Ref. 1, pp. 364–382), where Γ is a diagonal matrix of eigenvalues and U and W contain the corresponding eigenvectors. It is intended as a first step in Tucker and Messick's approach to an individual differences model for multidimensional scaling.[2]

As is, MATDEC will handle up to 100 variables or subjects, whichever is the lesser dimension of the data matrix. The program uses F4STAT (FORTRAN IV Statistical System) and, in particular, the routine SDGEXT to develop the characteristic roots and vectors of the crossproducts matrix. Further information about F4STAT can be obtained from Mr. Van Hassel, Educational Testing Service, (609) 921-9000, ext. 2557.

REFERENCES

1. Horst, P., *Matrix Algebra for Social Scientists*, (Holt, Rinehart & Winston, New York, 1963).

2. Tucker, L.R., and Messick, S., "An Individual Differences Model for Multidimensional Scaling," Psychometrika, *28*, 333–367 (1963).

000 0061

000 0061

USER INSTRUCTIONS

Input Deck

Title Card

Parameters	*Columns*	*Contents*
NV	1– 4	Number variables
N	5– 8	Number of subjects
IRC	9–12	1: input is rowwise 2: input is columnwise
IT	13–16	Tape unit containing input 5: input on cards
IR	17–20	1: input tape is to be rewound 0: input tape is not to be rewound

NOTE: Data matrix is dimensioned NV by N.

Variable–Format Card (optional: card input; IT=5)

Standard FORTRAN format card but delete word FORMAT.

Data Cards (optional)

Follow format on Variable-Format Card.

NOTE: If IRC=1 and an entire matrix row (N entries) is on a single Data Card, then the numbers of Data Cards = NV.

System Control Cards

JOB Card will be prepared by Educational Testing Service personnel.

System Card 1

Columns	*Contents*
1–36	//STEPNAME EXEC GITNGO,NAME=MATDEC

System Card 2

Columns	*Contents*
1–19	//GO.SYSIN DD *

System Card 3

Columns	*Contents*
1– 2	//

continued

Job Deck

JOB Card
System Card 1
System Card 2
Title Card

Parameter Card
Variable-Format Card (optional)
Data Cards (optional)

May be repeated for additional programs.

System Card 3

Description of Output

Printed Output: The parameters NV, N, IRC, IT, and IR are printed at the beginning of the job, and the title is printed at the top of each new page. The raw data are printed followed by the cross-products matrix. An abbreviated form of Γ (the diagonal elements only) is followed by U and W as labeled.

Tape Output: In addition, output will be written in binary on tape unit 3 in the following order.

(1) the 72–position array TITLE
(2) the diagonal elements of Γ

when NV $\leq$ N:

(3) the matrix U, rowwise
(4) the matrix W, columnwise

or, when NV > N:

(3) the matrix W, columnwise
(4) the matrix U, rowwise

If there are additional runs, FT03F002 (OS/360 file number; see contact person for further information) etc., will contain the subsequent output blocks. Note that U, Γ, and W are written in double precision.

SAMPLE INPUT

```
//JOBNAME  JOB (....
//STEPNAME EXEC GITNGO,NAME=MATDEC
//GO.SYSIN DD *
SAMPLE CASE TITLE LINE
   6   4   1   5
(4F1.0)
1649
5524
9452
4267
3155
7831
//
```

SAMPLE OUTPUT

```
        NO. OF VARIABLES =     6
        NO. OF SUBJECTS =     4
        IRC = 1
        INPUT TAPE =  5
        IR = 0   (1 MEANS REWIND)

ROW     1 OF DATA MATRIX

             1.00000      6.00000      4.00000      9.00000

ROW     2 OF DATA MATRIX

             5.00000      5.00000      2.00000      4.00000

ROW     3 OF DATA MATRIX

             9.00000      4.00000      5.00000      2.00000

ROW     4 OF DATA MATRIX

             4.00000      2.00000      6.00000      7.00000

ROW     5 OF DATA MATRIX

             3.00000      1.00000      5.00000      5.00000

ROW     6 OF DATA MATRIX

             7.00000      8.00000      3.00000      1.00000

CROSS-PRODUCTS

                  1            2            3            4

        1    181.00000    134.00000    119.00000     97.00000
        2    134.00000    146.00000     95.00000    109.00000
        3    119.00000     95.00000    115.00000    124.00000
        4     97.00000    109.00000    124.00000    176.00000



EIGENVALUES (GAMMA)

                  1

        1     22.25328
        2      9.30737
        3      5.82712
        4      1.48615

EIGENVECTORS (W')

                  1            2            3            4

        1      0.53954     -0.60364     -0.37722     -0.44969
        2      0.49038     -0.25521      0.78853      0.26949
        3      0.45743      0.16536     -0.48038      0.72983
        4      0.50910      0.73698      0.07187     -0.43876
```

continued

```
ROW     1 OF MATRIX U
             0.44458     0.55434     0.52843     0.09266

ROW     2 OF MATRIX U
             0.36403    -0.10912     0.23738    -0.80504

ROW     3 OF MATRIX U
             0.45489    -0.44619    -0.42886    -0.13300

ROW     4 OF MATRIX U
             0.42453     0.34661    -0.39660     0.03220

ROW     5 OF MATRIX U
             0.31194     0.26276    -0.40941     0.25284

ROW     6 OF MATRIX U
             0.43055    -0.54087     0.39443     0.51056
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $2.13.

Charge to user = computer costs + postage and handling + network overhead
= $2.13 + $5.00 + network overhead
= $7.13 + network overhead

CONTENTS—MATDEC

pages	
1	Identification & Abstract
3– 4	User Instructions
5– 7	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	Analysis of Change-over Experiments
CALLING NAME	ZFE-03, ZFE-04
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service (ETS)
AUTHOR(S) AND AFFILIATION(S)	Procedure: G. Beall formerly of Gillette Razor Co. Program: V. Halfmann formerly of ETS
LANGUAGE	FORTRAN
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Off. of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

The class of designs covered by this program is latin squares or Youden rectangles (incomplete latin squares). These may be repeated fully or in part. The design may be defective, i.e., certain whole rows may be missing, but no allowance has been made for missing cells, i.e., single observations.

The purpose of this program is primarily to analyze the results for effect of treatment with allowance for carry-over of preceding treatment. There is also direct testing of the significance of carry-over. There is included parallel estimation and testing of significance without allowance for carry-over. The program is self-contained and does not require any external subroutines, such as might be presumed to exist in one form or another at computation centers.

The program, as presently stored, allows analysis for designs up to 40 rows (or blocks). This is limited by the dimensions of the data matrices. It could be readily enough changed to 500 or 1000 if such an experiment were involved. The analysis has been contrived so that such change does not increase the size of the matrix involved in equation solving.

continued

Explanation

The basic equations are,

1. $y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk}$

where y_{ijk} is an observation assumed built of a general level μ, effect of the i^{th} row or individual α_i, the j^{th} period or column β_j, the k^{th} treatment γ_k and extraneous variability ε_{ijk}. This equation obtains for the first column or period when there has been no conditioning period. For the following periods,

2. $y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_\ell + \varepsilon_{ijk\ell}$

where the effect of the ℓ^{th} treatment in the preceding period is δ_ℓ. For a conditioned experiment Equ. (2) obtains in all columns.

The data actually considered are the differences within rows such as,

$$y_{ijk\ell} - y_{i'j'k'\ell'} = \beta_j - \beta_{j'} + \gamma_k - \gamma_{k'} + \delta_\ell - \delta_{\ell'}$$

$$(j' \neq j\ ,\ k' \neq k\ ,\ \ell' \neq \ell)\ .$$

These differences are then set forth in a matrix. Thus for an unconditioned latin square for which the first line is,

Design:	(1)	(2)	(4)	(3)
Result:	4	5	7	6

we may consider the two differences,

$$y_{111} - y_{1221} = \beta_1 - \beta_2 + \gamma_1 - \gamma_2 - \delta_1 + \varepsilon_{111} - \varepsilon_{1221}$$

$$y_{1221} - y_{1342} = \beta_2 - \beta_3 + \gamma_2 - \gamma_4 + \delta_1 - \delta_2 + \varepsilon_{1221} - \varepsilon_{1342}$$

which results in two lines of the matrix as follows:

β_1	β_2	β_3	β_4	γ_1	γ_2	γ_3	γ_4	δ_1	δ_2	δ_3	δ_4	Result
+1	–1			+1	–1			–1				–1
	+1	–1			+1		–1	+1	–1			–2

Least-squares equations are in the same form. For instance, to get the equation associated with β_2, each line is multiplied by its content in the β_2 column and the product accumulated over all columns. For each set of effects (β, γ or δ), the last equation is replaced by a condition equation,

continued

$$\sum_j \hat{\beta}_j = \sum \hat{\gamma}_k = \sum_\ell \hat{\delta}_\ell = 0 .$$

The analysis without carry-over (SANS DELTA) is gotten by replacing, temporarily, all equations appropriate to δ by $\hat{\delta}_\ell = 0$.

The analysis without treatment (SANS GAMMA) is obtained by the temporary replacement, $\hat{\gamma}_k = 0$.

The residual variability is gotten in several steps. First the variability residual on β , γ and δ is

$$3. \quad \sum_{ijk} y^2_{ijk} + \sum_{ijk\ell} y^2_{ijk\ell} - \sum_j \hat{\beta}_j (\sum_{ik} y_{ijk} + \sum_{ik\ell} y_{ijk\ell})$$

$$- \sum_k \hat{\gamma}_k (\sum_{ij} y_{ijk} + \sum_{ij\ell} y_{ijk\ell}) - \sum_\ell \hat{\delta}_\ell \sum_{ijk} y_{ijk\ell} .$$

Secondly, an estimate of $\hat{\mu}$ is made by finding from Equ. (1) or (2) the mean of the values

$$y'_{ijk} = y_{ijk} - \hat{\beta}_j - \hat{\gamma}_k$$

$$y'_{ijk\ell} = y_{ijk\ell} - \hat{\beta}_j - \hat{\gamma}_k - \hat{\delta}_\ell \quad .$$

Thirdly, estimates of $\hat{\alpha}_i$ are made by finding the mean of the values

$$y''_{ijk} = y_{ijk} - \hat{\mu}$$

$$y''_{ijk\ell} = y'_{ijk\ell} - \hat{\mu} .$$

The residual variability as from equation (3) is then further and finally reduced by

$$\hat{\mu} \sum_{ijk\ell} y_{ijk\ell} + \sum_i \alpha_i \sum_{jk\ell} y_{ijk\ell} .$$

Residual variability on the effects of $\hat{\mu}$, rows and columns only (SANS DELTA & SANS GAMMA) is gotten by the formula familiar in analysis of variance.

continued

The test of significance for treatments without allowance for carry-over is based on residual variability SANS DELTA less residual variability SANS DELTA & SANS GAMMA. The test with allowance for carry-over is from residual variability on FULL MATRIX less that on SANS GAMMA. The test for carry-over is from residual variability on FULL MATRIX less that on SANS DELTA.

It need only be added that there is incorporated a test on whether the situation is underdetermined. The program counts the number of different row patterns, multiples this number by the number of columns, and checks whether the result exceeds the number of independent parameters to be estimated. In the case of underdetermination, it refuses to analyze. A second type of refusal arises if the simultaneous equations prove insoluble, which may arise if the design is redundant. Finally, if there is no residual freedom, the program will estimate the parameters but declare F = 0.

USER INSTRUCTIONS

Because there are two experimental designs for change-over experiments, two programs have been written. The first, ZFE-03, analyzes the results from a design that did not include a conditioning week. The second, ZFE-04, analyzes the results from a design that did include a conditioning week. In each case the Parameter Card will give the number of treatments, the number of columns of result, the number of columns of treatment patterns, and the number of rows. In the unconditioned experiment the number of columns of results will equal the number of columns of treatment patterns. In the conditioned experiment, the number of columns of treatment patterns will be one greater than the number of columns of results.

Input

Parameter Card

Columns	*Contents*
1– 5	Total number of treatments possible
6–10	Number of columns in treatment pattern
11–15	Number of columns of results
16–20	Number of rows of treatments and results

Note: For an experiment with a conditioning period, the number of columns in the treatment pattern will be one greater than the number of columns of results. For an unconditioned experiment, the two numbers will be identical.

Row of Design Cards (one for each row)

Format	*Contents*
6I5	Statement of treatments

Row of Result Cards (one for each row)

Format	*Contents*
6F5.0	Statement of results

System Control Cards

JOB Card will be prepared by ETS personnel.

continued

System Card 1

Columns	*Contents*
1–30	//STEPNAME EXEC GITNGO,NAME=
31–36	ZFE-03 or ZFE-04 (see above)

System Card 2

Columns	*Contents*
1–19	//GO.SYSIN DD *

System Card 3

Columns	*Contents*
1– 2	//

Job Deck

JOB Card
System Card 1
System Card 2
Input Deck
System Card 3

Output

The design and results are printed out. The least squares estimates for columns and treatments are printed (SANS DELTA) as they are found without consideration of carry-over. Then the estimates for columns, treatments and carries-over are shown (FULL MATRIX). Finally, the analyses for significance are printed. There is a test for treatment, without allowance for carry-over, a test for treatment with allowance, and a test for carry-over. In each case there is shown the variance ratio, F, with its two statements of freedom.

SAMPLE INPUT

```
//                      JOB         (....
//STEPNAME  EXEC   GITNGO,NAME=ZFE-03
//GO.SYSIN  DD     *
      7     4       4  18
      1     2       4   7
      2     3       5   1
      4     5       7   3
      5     6       1   4
      6     7       2   5
      7     1       3   6
      2     6       4   3
      3     7       5   4
      4     1       6   5
      5     2       7   6
      6     3       1   7
      7     4       2   1
      1     2       4   7
      2     3       5   1
      3     4       6   2
      4     5       7   3
      6     7       2   5
    55.   37.     47. 45.
    37.   25.     25. 38.
    24.   24.     29. 46.
    36.   44.     39. 52.
    42.   54.     50. 55.
    49.   37.     56. 62.
    42.   45.     35. 32.
    55.   60.     52. 60.
    54.   60.     57. 54.
    42.   44.     46. 44.
    46.   48.     54. 47.
    42.   54.     42. 42.
    49.   42.     47. 48.
    34.   43.     46. 54.
    52.   55.     42. 42.
    38.   38.     26. 28.
    50.   62.     50. 61.
    42.   49.     51. 63.
//
```

SAMPLE OUTPUT

```
        ESTIMATES OF VARIABLES


SANS DELTA                                    FULL MATRIX


                          BETA
     -1.3858                                      -1.3742
     -0.1037                                      -0.1668
     -1.3409                                      -1.3384
      2.8304                                       2.8795
                          GAMMA
     -1.9907                                      -1.4136
     -1.1776                                      -0.5279
      4.1289                                       3.8279
     -0.6008                                      -0.9817
      0.1367                                      -0.6040
     -1.2811                                      -1.4894
      0.7846                                       1.1887
                          DELTA
     -0.0                                          0.7361
     -0.0                                          2.8368
     -0.0                                          0.7053
     -0.0                                         -0.1414
     -0.0                                         -2.0391
     -0.0                                         -1.4626
      0.0                                         -0.6351


          ANALYSIS

                  FACTORS                   | D.F. |  RESIDUAL VAR.
--------------------------------------------+------+---------------
MU ROWS COLUMNS                     CONTROL  | 51.  |     2188.37
CONTROL FACTORS PLUS      TREATMENTS        | 45.  |     1973.20
--------------------------------------------+------+---------------


   F        =  (     2188.37 -   1973.20 )/ 6.    =    0.82
    6.,45.            1973.20 / 45.


                  FACTORS                   | D.F. |  RESIDUAL VAR.
--------------------------------------------+------+---------------
MU ROWS COLUMNS CARRIES-OVER CONTROL        | 45.  |     2045.65
CONTROL FACTORS PLUS     TREATMENTS         | 39.  |     1874.19
--------------------------------------------+------+---------------


   F        =  (     2045.65 -   1874.19 )/ 6.    =    0.59
    6.,39.            1874.19 / 39.


                  FACTORS                   | D.F. |  RESIDUAL VAR.
--------------------------------------------+------+---------------
MU ROWS COLUMNS TREATMENTS     CONTROL      | 45.  |     1973.20
CONTROL FACTORS PLUS     CARRIES-OVER       | 39.  |     1874.19
--------------------------------------------+------+---------------


   F        =  (  1973.20 -   1874.19 )/ 6.    =    0.34
    6.,39.         1874.19 / 39.
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $3.80.

Charge to user = computer costs + postage and handling + network overhead
= $3.80 + $5.00 + network overhead
= $8.80 + network overhead

CONTENTS—ZFE-03, ZFE-04

pages

1– 4 Identification & Abstract
5– 6 User Instructions
7– 8 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Multivariate-Analysis of Variance
CALLING NAME	MANOVA
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service (ETS)
AUTHOR(S) AND AFFILIATION(S)	Elliot M. Cramer, ETS Amended by: Charles C. Hall, ETS
LANGUAGE	FORTRAN II
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Off. of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

MANOVA performs univariate and multivariate analyses of variance, of covariance, and of regression. It is quite general and, initially at least, runs should be coordinated with the Office of Computation Sciences through the contact person until the user becomes familiar with the preparation of parameter cards involved. It will perform univariate and multivariate analyses of variance with complete factorial designs or incomplete designs, and with or without covariates, discriminant analyses and canonical correlations. An exact solution in either the orthogonal or nonorthogonal case is provided, and options include single- or multiple-degree-of-freedom contrasts in the main effect or interactions, transformation of variables, and orthogonal polynomial contrasts with equally- or unequally-spaced points. Reanalyses may be done with different criteria, covariates, contrasts and models.

Limitations include a total of 40 variables, no more than 100 nonvacant cells, and up to eight factors with a maximum of 20 levels in each. There is no limitation on the number of subjects.

continued

REFERENCES

Bock, R.D., "Programming Univariate and Multivariate Analysis of Variance," Technometrics, 5, 95–117 (1963).

Clyde, D.J., Cramer, E.M., & Sherin, R., *Revised Manova Program* (The University of Miami Biometric Laboratory, Coral Gables, Florida, 1967).

Graybill, F.A., *An Introduction to Linear Statistical Models* (McGraw–Hill Book Co., Inc., New York, 1961) 223–253.

Morrison, D.F., *Multivariate Statistical Methods* (McGraw–Hill Book Co., Inc., New York, 1967) 159–206.

Rao, C.R., *Advanced Statistical Methods in Biometric Research* (John Wiley & Sons, Inc., New York, 1952) 236–272.

Roy, S.N., *Some Aspects of Multivariate Analysis* (John Wiley & Sons, Inc., New York, 1957) 76–86.

USER INSTRUCTIONS

Input

Input to the program consists of a variable number of cards depending upon the options used. The choice of the options is generally indicated by a 1 punch in a particular column of the Problem Card.

The types of cards used, identification, a summary description of contents and order of cards follow.

Title Card

Columns	*Contents*
1– 5	TITLE
6–80	Any title; multiple card may be used

Problem Card

Columns	*Contents* (b represents blank)
1– 4	PROB
6– 7	No. of variables ($\leq$ 40)
8	No. of factors ($\leq$ 8)
9	n: No. of Variable-Format Cards b: standard format
10	1: variable names supplied b: numbers used
11	1: subset of variables used as criteria b: all variables used
12	1: Significance Test Cards read b: standard ordering
13	1: last effect on Significance Test Cards obtained as residual and tested b: no test on residual effect
14	1: correlation between observed error variables and canonical variables printed b: correlation not printed
15	4: Input data on tape 4 b: Input data on tape 5

NOTE: Data on tape 5 *must* have LRL $\leq$ 70, RCT = 1, Block $\leq$ 70, Type 1. Tape labels *must* be bypassed by 7044 operator. (See contact person for further information.)

continued

Problem Card

Columns	*Contents* (b represents blank)
16	1: cell means not printed b: cell means printed
17	1: transformations of data b: no transformations
18	1: reduced model matrix printed b: matrix not printed
19	1: correlations among model parameters printed b: correlation not printed
20	1: reduced model matrix read in (Col. 12 = 1) b: matrix not read
21	1: mean omitted from model (Col. 12 = 1) b: mean included
22	1: sum of product matrix printed b: matrix not printed

Variable-Name Card (optional: PROB Col. (10) = 1)

Columns	*Contents*
1–10	Name of first variable
11–20	Name of second variable
⋮	⋮
71–80	Name of eighth variable Use number of names equal to number of variables.

Variable-Subset Card (optional: PROB Col. (11) = 1)

Columns	*Contents*
1– 2	Number of criteria (right-justified)
3– 4	Number of covariates (right-justified)
5–80	Of the V variables, punch variable numbers followed by covariates in 2–digit fields. Use additional cards if necessary starting in Col 1.

continued

Contrast Cards (entered for every factor in same order as factor codes)

Columns	*Contents* (b represents blank)
1	letter ID of factor (not W)
2	b: regular contrasts 1: special contrasts 2: orthogonal polynomial contrasts on following card 3: orthogonal comparison of i + first level minus average of previous levels
3–4	Levels for this factor (right-justified ≤ 20)
5–20	b: no partition of factor If partitioning to be done, punch degrees of freedom for each partition in 2–digit fields.
21–60	b: level codes are punched in Data Cards Level cards are punched in 2–digit fields otherwise.
61–80	Factor name

Special Contrast Cards (optional: Contrast Card Col. (2) = 1)

Columns	*Contents*
1–80	Square matrix with as many rows as levels in factor in 5–digit fields. Each row begins on new card; they must sum to zero.

(Contrast Card Col. (2) = 2)

Columns	*Contents*
1–80	For orthogonal polynomials, specify metric in 5–digit fields.

Significance Test Cards (optional: PROB Col. (12) = 1)

Columns	*Contents*
1–80	Code letters for effects of design separated by commas, followed by period. W denotes nesting. Ex: B,A,AB.

Variable-Format Cards (optional: PROB Col. (9) = n)

Columns	*Contents*
1–80	Variable format for one subject. Include level code for each factor and include parentheses.

continued

Transformation Card (optional: PROB Col. (17) = 1)

Columns	*Contents*
1 to number of variables	b: no transformation 1: square root transformation 2: log transformation 3: arcsin transformation 4–9: user's transformation; subroutine SPECTR must be changed

Data Cards (Format designated by PROB Col. (9))

Data must include: numbered level code(s) followed by scores. All Data Cards for one subject must be in order. Deck is followed by as many blank cards as there are Data Cards for one subject.

Reduced Model Matrix Cards (optional: PROB Col. (20) = 1)

First row of reduced model matrix punched in 5-column fields. Continue for remainder of rows in reduced model matrix.

Reanalysis Card (optional)

Columns	*Contents* (b represents blank)
1– 5	ANALY
6	b: all variables used as criteria 1: subset used as criteria
7	b: standard or diagonal ordering 1: Significance Test Cards read
8	b: previous contrasts used 1: new contrasts used
9	b: residual effect not tested 1: residual effect tested
10	b: reduced model matrix not read 1: reduced model matrix read
11	b: general mean included 1: general mean omitted
12	b: print complete statistical analysis 1: print estimates of effects only

continued

Variable-Subset Cards (optional: ANALY Col. (6) = 1)

Same format as previous Variable-Subset Cards

Contrast and Special Contrast Cards (optional: ANALY Col. (8) = 1)

Same format as previous Contrast Cards

Significance Test Cards (optional: ANALY Col. (7) = 1)

Same format as previous Significance Test Cards

Reduced Model Matrix Cards (optional: ANALY Col. (10) = 1)

Same format as previous Reduced Model Matrix Cards

Finish Card (placed at end of all problems)

Columns	*Contents*
1– 6	FINISH

System Control Cards

JOB Card will be prepared by ETS personnel.

System Card 1

Columns	*Contents*
1–36	//STEPNAME EXEC GITNGO,NAME=MANOVA

System Card 2

Columns	*Contents*
1–19	//GO.SYSIN DD *

System Card 3

Columns	*Contents*
1– 2	//

Job Deck

JOB Card
System Card 1
System Card 2
Input Deck
System Card 3

continued

Output

The output is dependent upon the specifications in the input cards. It may include

1. description and identification of variables and factors
2. means and standard deviations
3. reduced model matrix
4. correlations of effects
5. univariate analysis table (ANOVA)
6. multivariate

    (a) error correlations
    (b) estimates of effects tested against the error term
    (c) analysis of regression if there are covariates

7. a page with multivariate and univariate tests for each effect. In the program, sufficient diagnostic output is given to isolate an error. For more detail, consult the complete writeup (Ref. 1).

Restrictions

No. of variables ($\leq$ 40)
No. of factors ($\leq$ 8)
No. of levels for each factor ($\leq$ 20)
No restriction on number of subjects

REFERENCES

Clyde, D.J., Cramer, E.M., & Sherin, R., *Revised MANOVA Program* (The University of Miami Biometric Laboratory, Coral Gables, Florida, 1967).

SAMPLE INPUT

```
// JOB      (....
//STEPNAME  EXEC  GITNGO,NAME=MANOVA
//GO.SYSIN  DD    *
TITLE PROBLEM 1 FROM HALL AND CRAMER MANOVA PROGRAM WRITTEN IN FORTRAN II
TITLE 2X2 FACTORIAL DESIGN WITH 6 CRITERIA
TITLE REANALYZE WITH 1 CRITERION AND 5 COVARIATES
PROB  6211   1   11
        ONE         TWO     THREE       FOUR       FIVE       SIX
A  2                                                            ALPHA
B  2                                                            BETA
(2I1,10X,6F6.0)
115    41     569.  156.  104.  506.  4.     9.
115    42     475.  120.  105.  366.  4.     16.
115    43     641.  83.   82.   815.  4.     16.
115    44     779.  104.  104.  331.  4.     17.
115    45     587.  98.   53.   564.  3.     13.
115    46     841.  129.  71.   519.  5.     5.
115    47     907.  90.   49.   416.  3.     16.
115    48     698.  76.   53.   492.  4.     9.
115    49     849.  132.  89.   459.  5.     1.
115    50     505.  166.  207.  474.  4.     -9.
126    51     557.  91.   62.   513.  3.     26.
126    52     649.  114.  52.   416.  4.     16.
126    53     714.  81.   50.   491.  4.     0.
126    54     611.  125.  80.   630.  3.     25.
126    55     713.  84.   57.   471.  4.     4.
126    56     644.  97.   63.   453.  4.     5.
126    57     593.  83.   55.   546.  4.     8.
126    58     536.  125.  85.   364.  3.     24.
126    59     988.  109.  94.   554.  4.     7.
126    60     584.  120.  91.   503.  4.     4.
218    71     935.  72.   67.   623.  4.     12.
218    72     846.  96.   79.   539.  4.     2.
218    73     704.  109.  45.   355.  2.     7.
218    74     953.  142.  67.   432.  3.     22.
218    75     553.  97.   80.   495.  3.     16.
218    76     592.  82.   67.   362.  2.     17.
218    77     529.  85.   50.   657.  2.     14.
218    78     556.  99.   67.   589.  1.     18.
218    79     419.  103.  77.   452.  3.     15.
218    80     598.  78.   59.   397.  4.     0.
229    81     662.  75.   48.   385.  4.     14.
229    82     668.  88.   48.   361.  2.     8.
229    83     519.  110.  48.   391.  3.     11.
229    84     449.  90.   66.   484.  4.     9.
229    85     647.  80.   56.   482.  4.     2.
229    86     589.  64.   44.   337.  3.     15.
229    87     846.  73.   60.   598.  4.     6.
229    88     748.  96.   65.   601.  4.     10.
229    89     763.  135.  92.   480.  2.     12.
229    90     578.  102.  65.   683.  3.     -8.
                                                    <———— Blank Card
ANALY1
 1 5 5 1 2 3 4 6
/*
```

SAMPLE OUTPUT

```
MULTIVARIATE ANALYSIS OF VARIANCE
DATED 3/21/68

 PROBLEM 1 FROM HALL AND CRAMER MANOVA PROGRAM WRITTEN IN FORTRAN II
 2X2 FACTORIAL DESIGN WITH 6 CRITERIA
 REANALYZE WITH 1 CRITERION AND 5 COVARIATES

PROBLEM   1   6 VARIABLES  2 FACTORS

        ONE           TWO          THREE          FOUR          FIVE          SIX

 6 CRITERIA   0 COVARIATES  WITH THE FOLLOWING VARIABLES
        ONE           TWO          THREE          FOUR          FIVE          SIX

FACTOR A    2 LEVELS          ALPHA
  DEVIATION CONTRASTS

FACTOR B    2 LEVELS          BETA
  DEVIATION CONTRASTS

FORMAT OF DATA CARDS
   (2I1,10X,6F6.0)

  4 CELLS



                                         MEANS AND STANDARD DEVIATIONS


FACTOR                                                  VARIABLE
 A  B                                      ONE            TWO         THREE         FOUR         FIVE          SIX
 1  1          10 OBS
                              M         685.100       115.400        91.700      494.200        4.000        9.300
                              SD        153.122        30.497        46.094      133.073        0.667        8.367
 1  2          10 OBS
                              M         658.900       102.900        68.900      494.100        3.700       11.900
                              SD        130.096        17.760        15.868       74.866        0.483        9.927
 2  1          10 OBS
                              M         668.500        96.300        65.800      490.100        2.800       12.300
                              SD        183.563        19.889        11.717      108.459        1.033        7.134
 2  2          10 OBS
                              M         646.900        91.300        59.200      480.200        3.300        7.900
                              SD        119.088        20.769        14.172      116.118        0.823        6.757


COMPLETE FACTORIAL WITH NO MISSING CELLS


                              REDUCED MODEL MATRIX


FACTOR                                          PARAMETER
 A  B                                  1          2          3          4
 1  1                                1.000      1.000      1.000      1.000
 1  2                                1.000      1.000     -1.000     -1.000
 2  1                                1.000     -1.000      1.000     -1.000
 2  2                                1.000     -1.000     -1.000      1.000

CORRELATIONS OF CONTRASTS WITH STANDARD DEVIATIONS OF CONTRASTS DIVIDED BY STANDARD DEVIATIONS OF VARIABLES ON DIAGONAL


CONTRAST          1          2          3          4
   1            0.158
   2            0.000      0.158
   3            0.000     -0.000      0.158
   4            0.000     -0.000      0.000      0.158
```

continued

```
WITHIN CELLS CORRELATIONS OF CRITERIA WITH STANDARD DEVIATIONS ON DIAGONAL ADJUSTED FOR  0 COVARIATES


VARIABLE                    ONE         TWO        THREE        FOUR        FIVE        SIX
     ONE                 148.533
     TWO                  -0.063      22.762
   THREE                  -0.171       0.633      26.208
    FOUR                   0.080      -0.122       0.021     110.182
    FIVE                   0.295      -0.122       0.076       0.074       0.778
     SIX                  -0.097      -0.058      -0.261 ,    -0.117      -0.440       8.141



ESTIMATES ADJUSTED FOR  0 COVARIATES
                                              CRITERIA
CONTRAST                      ONE            TWO          THREE           FOUR           FIVE           SIX
A
     1                       7.150          7.675          8.900          4.500          0.400          0.250
      LAST
     2                      -7.150         -7.675         -8.900         -4.500         -0.400         -0.250
B
     1                      11.950          4.375          7.350          2.500         -0.050          0.450
      LAST
     2                     -11.950         -4.375         -7.350         -2.500          0.050         -0.450
AB
     1                       1.150          1.875          4.050         -2.450          0.200         -1.750
      LAST
     2                      -1.150         -1.875         -4.050          2.450         -0.200          1.750



TEST OF AB

TESTS OF SIGNIFICANCE USING WILKS LAMBDA CRITERION   AND CANONICAL CORRELATIONS
TEST OF ROOTS              F          DFHYP       DFERR     P LESS THAN      R
 1 THROUGH  1            0.557        6.000      31.000        0.761        0.312


                    UNIVARIATE F TESTS                   STANDARDIZED DISCRIMINANT FUNCTION COEFFICIENTS
VARIABLE         F( 1,    36)     MEAN SQ     P LESS THAN     1
     ONE             0.002        52.899         0.961      -0.167
     TWO             0.271       140.625         0.606       0.169
   THREE             0.955       656.099         0.335       0.222
    FOUR             0.020       240.101         0.889      -0.135
    FIVE             2.642         1.600         0.113       0.744
     SIX             1.848       122.500         0.182      -0.327




TEST OF B

TESTS OF SIGNIFICANCE USING WILKS LAMBDA CRITERION   AND CANONICAL CORRELATIONS
TEST OF ROOTS              F          DFHYP       DFERR     P LESS THAN      R
 1 THROUGH  1            0.726        6.000      31.000        0.632        0.351


                    UNIVARIATE F TESTS                   STANDARDIZED DISCRIMINANT FUNCTION COEFFICIENTS
VARIABLE         F( 1,    36)     MEAN SQ     P LESS THAN     1
     ONE             0.259      5712.089         0.614       0.510
     TWO             1.478       765.624         0.232      -0.098
   THREE             3.146      2160.899         0.085       1.052
    FOUR             0.021       249.998         0.887       0.051
    FIVE             0.165         0.100         0.687      -0.267
     SIX             0.122         8.100         0.729       0.362
```

continued

```
TEST OF A

TESTS OF SIGNIFICANCE USING WILKS LAMBDA CRITERION   AND CANONICAL CORRELATIONS
TEST OF ROOTS                  F           DFHYP        DFERR     P LESS THAN      R
 1 THROUGH  1               3.286          6.000        31.000       0.013        0.624


                       UNIVARIATE F TESTS                   STANDARDIZED DISCRIMINANT FUNCTION COEFFICIENTS
VARIABLE           F( 1,   36)     MEAN SQ     P LESS THAN     1
      ONE               0.093     2044.853        0.763      -0.123
      TWO               4.648     2356.224        0.040       0.487
    THREE               4.613     3168.399        0.039       0.188
     FOUR               0.067      809.957        0.798       0.110
     FIVE              10.569        6.400        0.002       0.998
      SIX               0.038        2.500        0.847       0.558


DISCRIMINANT SCORES
CONTRAST                                                       1
    1                                                        0.757


CORRELATIONS BETWEEN VARIABLES AND COMPOSITE SCORES
VARIABLE                                                       1
      ONE                                                    0.064
      TWO                                                    0.446
    THREE                                                    0.449
     FOUR                                                    0.054
     FIVE                                                    0.679
      SIX                                                    0.041


MULTIVARIATE ANALYSIS OF VARIANCE


PROBLEM    1 REANALYSIS    1 WITH THE FOLLOWING    1 CRITERIA AND    5 COVARIATES
          FIVE            ONE            TWO          THREE          FOUR           SIX



SOURCE                              SS        DF          MS               F              P LESS THAN

WITHIN CELLS                      15.453      31         0.498
REGRESSION                         6.347       5         1.269           2.546             0.048
A                                  5.849       1         5.849          11.734             0.002
B                                  0.097       1         0.097           0.195             0.662
AB                                 0.614       1         0.614           1.232             0.276



ESTIMATES ADJUSTED FOR  5 COVARIATES
                                                  CRITERIA
CONTRAST                       FIVE
A
      1                       0.417
       LAST
      2                      -0.417
B
      1                      -0.052
       LAST
      2                       0.052
AB
      1                       0.128
       LAST
      2                      -0.128


RAW REGRESSION COEFFICIENTS

COVARIATES              WITHIN CELLS
        ONE                  0.001
        TWO                 -0.008
      THREE                  0.005
       FOUR                 -0.000
        SIX                 -0.037
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $5.28.

Charge to user = computer costs + postage and handling + network overhead
= $5.28 + $5.00 + network overhead
= $10.28 + network overhead

CONTENTS—MANOVA

pages

1– 2 Identification & Abstract
3– 8 User Instructions
9–12 I/O
13 Cost—Contents

DESCRIPTIVE TITLE	Structure-Factor Least-Squares Refinement Program for IBM 7090
CALLING NAME	ORFLS-PX
INSTALLATION NAME	University of Pittsburgh The Computer Center
AUTHOR(S) AND AFFILIATION(S)	Dr. Ryonosuke Shiono Crystallography Laboratory University of Pittsburgh
LANGUAGE	FORTRAN II, except SMI, which is in FAP(UMAP).
COMPUTER	IBM 7090 with 32K core storage 2–3 magnetic-tape units in addition to system input/output.
PROGRAM AVAILABILITY	Proprietary; usage permitted, but program deck or listing not available
CONTACT	Dr. Ryonosuke Shiono, Crystallography Laboratory, 300 Thaw Hall, University of Pittsburgh, Pittsburgh, Pa. 15213 Tel.: (412) 621-3500 ext. 7124

FUNCTIONAL ABSTRACT

ORFLS-PX is an extensively modified version of the FORTRAN least-squares program by Busing, Martin, and Levy. Their program,

> performs successive cycles of refinement using the full matrix of the normal equations. The parameters which may be adjusted include several scale factors, an overall temperature factor coefficient, the neutron scattering factors, individual atom multipliers, atomic coorinates, and isotropic or anisotropic individual atom temperature factor coefficients. The parameters to be varied may be specified arbitrarily, and structures of any symmetry may be accommodated. The refinement may be based either on the structure factors or their squares, and the observations may be weighted individually, or the use of unit weights may be specified.[1]

Features of ORFLS-PX

1. Reflexion-data input may be from cards or from magnetic tape

continued

2. Uses atomic-scattering-factor table of f versus sinθ
3. External scale factors now refer to F_{obs}, rather than to F_{cal}'s
4. Weighting scheme may be changed or weights modified by a subroutine
5. Limitations on maximum number of atoms, parameters to be used; number of parameters to be refined have been enlarged
6. Isotropic and anisotropic temperature factors for atoms may be used in mixed fashion
7. Anomalous dispersion correction may be made[2]
8. Calculated structure factors, together with the scales F_{obs} are written on a binary tape, which may be used as an input for the Fourier program or for a new cycle of the least-squares program.[3]
9. Using an output tape as the input, a part of the structure may be added as fixed contribution for a partial-mode calculation, (this either saves time in calculation or extends the limit of number of atoms to be used)
10. Different set of parameters may be refined in successive cycles of full-matrix refinement, when not all the parameters can be refined at the same time

General Description of the Program Setup

To provide more storage space for the parameter refinement, ORFLS-PX is divided into three separate core-load packages: MAIN(1), MAIN(2), and MAIN(3).

MAIN(1) reads the reflexion data on cards or on tape, interpolates the atomic-scattering factors for each kind, and assigns the weight according to the specification. This part may be bypassed if a tape from a previous cycle of calculation is used as the input. Otherwise, the program produces an intermediate output tape.

MAIN(2) reads back the MAIN(1) tape, one reflexion at a time and calculates structure factor and the necessary derivatives for least-squares refinement. The calculated structure factors and scaled F_{obs}' are on the output binary tape. This tape is rewound at the end of every cycle so that the data on this tape are the result of latest calculation. This tape may be used as the input for fixed contribution in a partial-mode calculation.

MAIN(3) prints out the correlation matrix and other information on various agreements at the end of all calculation.

continued

Card formats used for reflexion data and atomic parameters and for symmetry operations are identical to many other interconnected programs on the IBM 7090 and IBM 1130 as well as the IBM 1620. Usually, the user has only to punch a few Control Cards to run any one of the programs.

The Data and Parameter Cards are identical to those in the Block-Diagonal Least-Squares programs[3] on the IBM 7090 and 1130, and so the user may easily switch from one program to another.

The reflexion-data file on magnetic tape has, again, the same format between this and Block-Diagonal, E, Fourier and other programs on the IBM 7090 (see Ref. 3), and it may be used interchangeably.

ORFLS-PX is an independent program, as compared to some others incorporated in a big system, permitting one to write a subroutine to fit a particular problem, and replace the dummy variables provided in the program by one's own.

REFERENCES

1. Busing, W.R., Martin, K.O., and Levy, H.A., "A FORTRAN Crystallographic Least-Squares Program," ORNL Tech. Mem. No. TM305 (Oak Ridge National Laboratories, Oak Ridge, 1962). The modified version of this program is used for calculations that involve scattering data and crystallographic structures. Additional information can be obtained through the contact person.
2. Ibers, J.A., and Hamilton, W.C., "Dispersion Correction and Crystal Structure Refinement," Acta Cryst. *17*, (1964), 781-782.
3. Additional information can be obtained through the contact person.

USER INSTRUCTIONS

Note for Punching Cards for the Michigan System

The input routine used in FORTRAN in the Michigan System has one bad feature insofar as the standard IBM FORTRAN users are concerned. The I/O routine ignores any blanks in numeric field, whereas the IBM version considers these as zero. As a result, if the following zeros are not punched in an integer field, or in a floating-point number without a decimal point, the Michigan program will take the last punched digit (rather than the last column of the specified field) as the last digit of the field. Therefore, care must be taken to insure that all the cards are punched for the system.

Recommendation: Always punch decimal point in a floating-point number field.

Card Format

Title Card

Any Hollerith characters (alphanumeric) may be punched on Cols. 1–72. The information punched here will be printed on the output as heading.

Control Card No. 1 (5I5, 6F8.4)

Parameter	*Columns*	*Contents*
JB	5	0: Input reflexion data from cards 1: Input reflexion data on tape 9. An output tape of the data-reduction program, E program, or that of the least-squares program (this program or block-diagonal program) may be used. The program uses only h, k, ℓ indices, F_{obs}, σ, ID from the input tape, so that this mode may be used when an input tape contains no atomic f's (data-reduction or E-program output) or some change in weight or atomic f's is desired 2: Input reflexion on cards. Run only the Part I to produce an intermediate output tape (Tape 10) for use in Patterson synthesis

continued

Parameter	*Columns*	*Contents*
		3: Input reflexion data on tape from a previous least-squares calculation. The tape should be mounted on tape 10. All information on the tape is used without change, the program bypasses Part I, and the Cell-Constant Card, the Atomic-Scattering Cards, the Reflexion-Data Cards, and the remainder of this card are irrelevant
JA	10	0: Refine F_{obs} 1: Refine on $(F_{obs})^2$. This causes $(F_{obs})^2$ to be written on tape 10. Do not use this mode if F_{obs}'s are already squared
JW	15	0: Standard error σ's are punched on the Data Cards and/or they will be assigned by the subroutine MODIFY 1: Hughes' weighting scheme. When specified, F_{min} (*not* $4F_{min}$) value should be punched as A in Cols. 26–33 2: Cruickshank's weighting scheme. A, B, C values below should be supplied. 3: $\omega = 1/(\sin\theta/\lambda)$ 4: Constant weight, $\omega = 1.0$
JC	20	0: Subroutine MODIFY is not used 1: Enter subroutine MODIFY
JD	25	0: Normal 1: Set $\omega = 0.0$ for unobserved reflexion with ID = 1
A	26–33	F_{min}, value for Hughes' or A for Cruickshank's scheme. This may be used for any other purpose in your MODIFY subroutine
B	34–41	For Cruickshank's B or blank
C	42–49	For Cruickshank's C or blank
D	50–57	For use in MODIFY subroutine
E	58–65	
F	66–73	

continued

Weight ω is taken as $\omega = 1/(\sigma)^2$. If $\sigma = 0$, then the program sets $\omega = 0$. What is punched on the Reflexion-Data Cards or calculate according to some scheme and write on the tape is σ and not ω. F_{obs} values used here are those on the input cards, or on the tape 9, and they are not affected by the scale factors that the user provides.

Hughes' Scheme (Ref. 1)

$\omega = 1/|F_o|^2$, if $F_o \geq 4F_{min}$;

$\omega = 1/|4F_{min}|^2$, if $F_o < 4F_{min}$;

$\omega = 0$, if $F_o = 0$.

Cruickshank's Scheme (Ref. 2)

$\omega = 1/(A + BF_o + CF_o^2)$.

Cell Constant Card (3F7.3, 4F7.4)

Columns	*Contents*
1– 7	a
8–14	b
15–21	c
22–28	$\cos\alpha$
29–35	$\cos\beta$
36–42	$\cos\gamma$
43–49	λ (wavelength used)

a, b, c, and λ are in angstrom units.

Atomic-Scattering-Factor Cards (7F10.4)

This is the table of f versus $\sin\theta$, eight cards per atomic kind (51 entries of f). Atomic f's are punched in increasing order of corresponding $\sin\theta$ value[3] from 0.00 to 1.00 with an increment of 0.02. A 1620 FORTRAN program or similar one on the 1130 computer prepares the cards from the literature value. The order of sets for each atomic kind determines the number to be used on the parameter cards. Up to ten sets. A blank card terminates the atomic f cards.

continued

Reflexion-Data Cards (3I9, 2F9.2, F9.0, I5, 2F6.3)

The output of the 1130 data-processing program is in this format.

Parameter	*Columns*	*Contents*	
	1– 9	h	indices
	10–18	k	
	19–27	ℓ	
	28–36	F_{obs}	
	37–45	σ, if used, or blank	
	53–54	Scale-factor designation. 1-20. If blank, 1.0 is assumed	
	59	0: Observed reflexion 1: Unobserved reflexion	
RA	60–65	Extra fields for any desired values if needed	
RB	66–71	The values will be transferred to the input and the output tape of the refinement part and available in the calculation. Blank if not used	

A blank card terminates the reflexion data.

Control Card No. 2 (I6I5)

Parameter	*Columns*	*Contents*
NC	5	The number of cycles to be run. If NC = 0, a set of structure factors will be calculated but no refinement will be made
NA	8–10	The number of atoms in the asymmetric unit, which are used in the parameter cards
NQ	14–15	The number of scale factors used and appearing in the parameter list
ITF	20	0: Isotropic temperature factors and anisotropic temperature factors, or mixed 1: Conversion on the parameter cards of isotropic temperature factors to anisotropic mode before refining. If some of the atoms have already anisotropic temperature factors, this will not affect them.

continued

Parameter	*Columns*	*Contents*
NV	23–25	Number of parameters to be varied and refined. If NC = 0 (unrefined structure factors are used), then NV is irrelevant.
NAS	29–30	Number of anomalous scattering atoms in the parameter cards to apply the correction. The parameter cards for those atoms must be grouped and placed in front of other atoms.
NPT	35	0: normal calculation 1: Partial-mode calculation. Ac, Bc are taken from the input tape and added to the current calculated values. This assumes that an output tape of previous calculations is mounted on tape 10 for use as input data. Any atom with anomalous dispersion correction should be in the new part and may not be in the input tape.
NPRN	40	0: Print structure factors for every cycle. 1: Print only the final structure factors and suppress the printout of other cycles.
NR	45	0: Normal calculation with no parameter-selection change 1: First run of repeat calculations with change in Parameter-Selection Cards. This makes the program go back and read a new set of Control Cards. 2: Second and more run of the above cycle. The program will go back and read more Control Cards 3: Final run of the recycling. The program terminates at the end of this specification.
NRW	50	0: Normal 1: Copy back the final output results on tape 4 to tape 10. For this, tape 10 must have the protector ring in. This copying takes place only when the final calculation has been run successfully.

continued

Parameter	*Columns*	*Contents*
ISTP	55	0: Program will be terminated, if a nonpositive definite temperature factor results. 1: Program will proceed in the above case by resetting the temperature factor to 0.0001 (for isotropic) or the original value (for anisotropic).

Control Card No. 3 (6I5)

Parameter	*Columns*	*Contents*
NF	5	The number of different x-ray form factors. NF = 0 for neutron diffraction
ICENT	10	1: Centrosymmetric 2: Noncentrosymmetric
NS	14–15	The number of Symmetry Cards. For centrosymmetric structure, NS is equal to one-half the number of equivalent positions. The original x, y, z should be included. ($1 \leq NS \leq 24$)
IFSQ	20	1: Refine (scale x F) 2: Refine (scale x F)^2
IXFE	25	0: No extra tape for ORFFE[3] 1: Extra tape for ORFFE. This may not be specified if the input tape 9 is used. 2: Punch the same on cards for ORFFE.

Symmetry Cards (F11.6, 2I2, F11.6, 2I2, F11.6, 2I2)

For a centrosymmetric structure, those positions related by the center of symmetry are not required. The basic x, y, z position must be included. For a centered-lattice, those symmetries related by pure translations are not required, but the atom multiplier should be increased.

0	blank
x	1
-x	-1
y	2
-y	-2
z	3
-z	-3

continued

Parameters	*Columns*	*Contents*
	1–11	translational part of x
	12–13 14–15	symmetry for x direction
	16–26	translational part of y
	27–28 29–30	symmetry for y direction
	31–41	translational part of z
	42–43 44–45	symmetry for z direction

Scale-Factor Cards (8F9.6)

Parameters	*Columns*	*Contents*
	1– 9	first scale factor S_1
	10–18	second scale factor S_2

If there are more than eight scale factors, continue to additional cards. All scale factors in this program apply to the F_{obs} value, rather than to the F_{cal} values. When an output tape of a previous calculation is used as the input, F_{obs} values were already scaled so that the scale factors used here should be 1.0.

Atomic-Parameter Cards

a) First Card (A6, 3X, 5F9.6, 2F6.2)

Parameters	*Columns*	*Contents*
	1– 6	Any 6 Hollerith characters identifying atom i
	10–18	X-ray scattering factor identifier, to specify which atomic f to use. (1.0 ~ 10.0) For neutron problem, this is the neutron-scattering factor itself.
	19–27	A multiplier, a_i, this number is usually 1.0, except for a special position or disordered atom, or for a centered-lattice.
	28–36	x_i, fractional coordinate
	37–45	y_i
	46–54	z_i

continued

Columns	*Contents*	
55–60	$\Delta f'$ $\Delta f''$	if anomalous dispersion correction is applied, the values are taken as constant for $\sin\theta$ value.

b) Second Card (6F9.6)

Columns	*Contents*
1– 9	T_i, the isotropic temperature factor, or β_{11}
10–18	β_{22}, (if the field for β_{22} is zero, the program assumes isotropic temperature factor
19–27	β_{33}
28–36	β_{12}
37–45	β_{13}
46–54	β_{23}

The above Atomic Parameter Cards may be prepared from the SFII Coordinate Cards (7070 or 1620) by a 1620 program,[3] and also may be converted back to SFII form. If anomalous dispersion correction is applied for some atoms, the Parameter Cards for those atoms should be placed first.

Parameter-Selection Cards (7211)

These cards specify the NV parameters to be varied, each column of the cards for each parameter, in the order,

Columns	*Contents*		
1	NQ scale factors		
2	scattering factor multiplier	or	scattering factor multiplier
3	x_i		x_i
4	y_i		y_i
5	z_i		z_i
6	B_i		β_{11}
7			β_{22}
8			β_{33}
9			β_{12}

continued

Columns	*Contents*
10	β_{13}
11	β_{23}
⋮	⋮

Punch 0 for holding constant
1 for refinement

The set of six or 11 columns repeat for all atoms used. Continue from Col. 72 of one card to Col. 1 of a next card. At present the total number of parameters to be used including those not to be varied is 811. This card is not used if the number of cycle is zero (unrefined structure factors only).

The number of parameters to be specified for each atom is either six or 11, depending on whether the temperature factor of an atom is isotropic or anisotropic. When using mixed mode of temperature factors, that is, some atoms have isotropic and others have anisotropic temperature factors, care must be taken to count and fill up the columns properly for six or 11. There should be no gap between atoms, and the order of atoms here are identical to the order in which the atomic Parameter Cards are read.

Please note that these Parameter Selection Cards are different from the corresponding ones used for the Block-Diagonal Least-Squares programs[3] (7090, 1130) by Shiono.

Assembly of Cards

7090 ID Card (Name, time estimate, etc.)

$$ Comment card(s) to specify scratch or private tapes on 10 and 4.

$NEEDS 10

$HALT (If a private tape is used)

$REMOVE TAPES (If a private tape is used with ring on)

$EXECUTE,I/O DUMP

Program deck (binary cards)

$DATA

Title Card

Control Card No. 1

Cell Constant Card

continued

Atomic-Scattering-Factor Cards (Set of eight, as many sets as necessary up to ten. The order of f's corresponds to the specification on parameter cards.)

Blank Card (to terminate the atomic f's)

Reflexion-Data Cards (as many as necessary)

Blank Card (to terminate the reflexions)

Control Card No. 2

Control Card No. 3

Symmetry Card(s)

Scale Factor Card(s)

Atomic-Parameter Cards (pair of cards for each atom)

Parameter-Selection Card(s) (not used if unrefined structure factors desired)

If JB = 3, Control Card No. 2 follows immediately after Control Card No. 1, and the cards between them are not used.

If JB = 1, omit the Reflexion-Data Cards and a blank card behind them. Control Card No. 2 follows the blank card behind the atomic f's.

If JB = 1, mount the input tape on Tape 9, and a scratch tape on Tape 10. If JB = 3, mount the input tape on Tape 10. If JB = 2, the output is on Tape 10.

The calculated results will be written on Tape 4, and this may be used as an input to the Fourier program or as an input to a later cycle of least-squares program.[3]

When starting with the reflexion on cards, tapes 10 and 4 must have their protector rings on.

Refinement Cycles With Different Set of Parameters Varied

Since the total number of parameters to be varied and refined at a time is limited (at present 181) by the size of core storage, sometimes it is necessary to refine a part of the total structure in a cycle or two and then switch the part to be refined until all the desired parameters have been refined. This may be done by punching a 1, 2 or 3 in Col. 45 of the Control Card No. 2.

Example of a setup of cards is shown below.

continued

Program deck

$DATA

First set of Control Cards for the first selection, 1 in Control Card No. 2, Col. 45.

For second or more refinement cycles,

Title Card
Control Card No. 1 JB = 3 (Col. 5)
Control Card No. 2 NR = 2 (Col. 48)
Control Card No. 3
Symmetry Cards
Parameter Selection Cards

Repeat above set of cards as many as necessary. For the final set of parameter selection (this may follow the first set immediately),

Title Card
Control Card No. 1 JB = 3 (Col. 5)
Control Card No. 2 NR = 3 (Col. 45)
Control Card No. 3
Symmetry Cards
Parameter Selection Cards

Symmetry Cards are usually redundant, but have to be present each time.

If certain part of the total structure is not going to be refined in all cycles, it will be faster to have this part calculated first and use the output tape as the input for fixed contribution.

Program Deck Makeup

MAIN(1)
SCAT
SBTP
MODIFY (Dummy)
BREAK CARD
MAIN(2)
CALC
TEST
SMI
SBTOP
PTOSB
DTOD

continued

```
OLDIOH
PATCH (Dummy)
RESETX (Dummy)
RESETB (Dummy), RSETW
BREAK CARD
MAIN(3)
BREAK CARD
```

A BREAK CARD is recognized as a card with 12, 7 and 9 punches on Col. 1 and rest of the columns all blank up to Col. 72.

Magnetic Tape Assignment

Logical tape addresses used in the program as compiled, are as follows.

Symbolic	*Logical*	*Physical*	
NTPA	7	B1	Michigan System Input
NTPB	6	A3	Michigan System Output (print)
NTPF	5	A3	Michigan System Output (card)
NTPG	10	A5	Output of Part I, Input of Part II
NTPC	9	B5	Tape input data to Part I (if used) or output tape of parameters to be used in ORFFE.[3]
NTPE	4	A4	Output of Part II (identical format of NTPG)

Specification of the Subroutines for the User

1. MODIFY Subroutine

This subroutine is entered, if so specified, in the Part I, to modify reflexion data. After reading a reflexion from card or tape, the main program calculates the Bragg angle and the weight according to the specification and enters this subroutine, just before writing the reflexion on another tape,

```
     SUBROUTINE MODIFY (FH,FK,FL,FOBS, FCAL, AΦ,AC,BΦ,BC,SIG,
                        Q,ID,SINT,CA,CB,CC,CD,CE,CF,RA,RB)
```

Parameters	*Contents*
FH,FK,FL	Miller indices h,k,ℓ

continued

Parameters	*Contents*
FOBS	observed F (or F^2), punched on card or from Tape 9
FCAL	calculated F
AΦ	observed F, real part
AC	calculated F, real part
BΦ	observed F, imaginary part
BC	calculated F, imaginary part
SIG	standard error of F
Q	designation for scale factor
ID	0 or 1 for observed or unobserved
SINT	sine theta (Bragg angle)
CA,CB,etc.	A,B,C,D,E,F punched on the Control Card
RA,RB	Extra values on reflexion data on tape. These values may be entered from the input card. Partial difference Fourier requires A_{cal}, B_{cal} (partial) be transferred to RA,RB here. (see Fourier writeup)[3]

The values such as FCAL, AΦ, AC, BΦ, BC are meaningful only when the input is the output tape of a previous structure factor (or least-squares) calculation.

In order to exclude a reflexion from the least-squares refinement, SIG should be set to 0.0, which will in turn set ω (weight) = 0. Should a reflexion be omitted entirely from the rest of the processing, set ID = 10.

The following three subroutines are explained in Ref. 4 and the user should consult that report. Only the dimension of some arrays have been changed in the current program and they are listed here.

```
    SUBROUTINE PATCH (I,TJ,HJ,HHJ)
      DIMENSION HJ(3), HHJ(6)

    SUBROUTINE RESETX (XYZ)
      DIMENSION XYZ (3,85)

    SUBROUTINE RESETB (BETA)
      DIMENSION BETA (6,85)
```

continued

1. RSETW Subroutine

This subroutine is entered for each reflexion after the completion of the structure factor calculation and before the weight is multiplied to various values in derivative calculation.

```
SUBROUTINE RSETW (FOBS,FCAL,A,B,SIGYO,SINT,ID)
RETURN
END
```

FOBS	F_{obs} value already scaled
FCAL	F_{cal} (absolute)
A,B	A,B part of F_{cal} with signs
SIGYO	σ, for zero weight, set $\sigma = 0$
SINT	$\sin\theta$ value
ID	0 for observed and 1 for unobserved

Any changes made here will not affect the contents of Input Tape 10 unless it is so specified to copy back the final Output Tape 4 to Tape 10.

When the user is supplying his subroutine to replace a dummy subroutine in the binary deck, this may be done in two ways,

(a) Compile and punch out the object deck of the subroutine first, and then place this binary deck in front of the dummy subroutine which is going to be replaced, but not before the BREAK Card preceding that part of the program. The system will use the first subroutine loaded.

(b) You may compile a subroutine and execute the entire program at the same time. For this, place the source deck instead of the binary deck of the subroutine as in (a), with a card $COMPILE FORTRAN, PUNCH OBJECT, PRINT OBJECT preceding it. The system first compiles the subroutine and then executes the entire program.

In either case, the cards for MODIFY should be placed in front of the Binary Deck, and those for PATCH, RESETX, RESETB or RSETW should be placed immediately behind the first BREAK Card. If the compilation is done with the execution, Tape 9 is needed in compilation and therefore, cannot be used as the input tape.

Atoms in Special Positions

Atoms in special positions have to be handled differently from other atoms in general positions. Three subroutine entries are

continued

provided to cope with the situation, if necessary, and they are PATCH, RESETX, and RESETB. The dummy subroutine provided with the deck may be replaced by a user's subroutine, by placing the source statement cards preceded by a $COMPILE FORTRAN Card, or the compiled Binary Cards in front of the one to be replaced.

The detailed instructions for handling special positions are given in Ref. 4, pp. 14–22.

Reflexion-Tape Record Format

The program uses the identical record format for the intermediate-input-reflexion data, and the output of structure-factor calculation. This format is further identical to that of the Fourier Input Tape, so that either the intermediate tape (Tape 10) or the final output (Tape 4) may be used as an input of the Fourier program[3] for calculating Patterson summation or Fourier summation, respectively.

The tape is written in binary mode, and the first record is different from the rest of the data.

First Record (13 words)

$1/4 \cdot a^{*2}$, $1/4 \cdot b^{*2}$, $1/4 \cdot c^{*2}$, $1/4 \cdot a^{*} b^{*} \cos\gamma^{*}$, $1/4 \cdot a^{*} c^{*} \cos\beta^{*}$, $1/4 \cdot b^{*} c^{*} \cos\alpha^{*}$, a, b, c, $\cos\alpha$, $\cos\beta$, $\cos\gamma$, λ.

Second to the (last–1) Record (26 words)

h, k, ℓ, F_{obs}, F_{cal}, A_{obs}, A_{cal}, B_{obs}, B_{cal}, σ, Q, ID, $(\sin\theta/\lambda)^2$, $\sin\theta$, (f_i, i=1, 10), RA, RB.

Parameters	*Contents*
h,k,ℓ	Miller indices
σ	Standard error
Q	Scale-factor identifier
ID	0 or 1 for observed or unobserved
RA,RB	Extra values

Only ID is in fixed-point number (integer).

Last Record

Identical to the preceding reflexion data except h = 999. This is used to recognize the end of data.

continued

Programmed Termination

In several points of the program execution, the run will be terminated before completion when certain conditions are not met.

1. Parameter Selection Cards Error

   If the number of parameters to be refined does not match the sum of 1's punched on the Parameter Selection Cards, the program will exit.

2. Nonpositive-Definite Temperature Factor

   If a temperature factor becomes nonpositive-definite after applying a shift, the program will terminate any further refinement (Reason 1).

3. No Improvement in Refinement

   If $\Sigma\omega(\Delta F)^2$/(No. of observation-No. of Parameters) is not decreasing from the previous cycle, the refinement will be terminated at the end of current cycle (Reason 2).

4. Singular Matrix

   If the matrix to solve the shifts is a singular matrix, the refinement will be terminated. When this happens, the matrix elements printed may be partially inverted.

Limitations

The program deck as compiled has the following capacity for various variables.

Variables	*Maximum Limit*	
Number of reflexions	no limit (one magnetic tape)	
Number of atoms	85	85
Number of scale factors	20	20
Number of total parameters	811	811
Number of parameters to be refined	181	181
Number of symmetry relations	24	24
Number of different kind of atoms	10	10
Number of anomalous dispersing atoms	10	10

The part 2 of the program has some locations available for the use of user's subroutines PATCH, RESETX, RESETB and RSETW.

continued

When a structure has some variables beyond the limit of the numbers listed above, it may be possible to change the program to handle the structure by recompiling the program with changed dimension statements.

With the partial mode, unlimited number of atoms may be used with multiple passes.

Output

I. Card

New scale factors, and the atomic parameters will be punched out for every cycle of refinement. These cards are in the format identical to the input, and may be used in further refinement. If so specified, the input cards for ORFFE[3] may be punched.

II. Print

All information on the two Control Cards (Nos. 2 and 3) as well as the Parameter Cards will be printed first. For each cycle, all structure factors, the agreements and the old and new parameters will be printed. Also one more set of structure factors will be calculated and printed out using the last parameters. If intermediate printing is suppressed, structure factors are printed only from the last calculation.

At the end, the correlation matrix for the parameter, agreement indices for various groups of reflexions and average $(\Delta F/k\sigma)^2$ will be printed out.

NOTE: If a private tape is used for Tape 10 and/or 4, a card with $HALT punched in Cols. 1–5 should be inserted immediately behind the $$ Card(s) for comment. This will halt the program and enable operator to mount the tape.

REFERENCES

1. Hughes, E.W., "The Crystal Structure of Melamine," J. Amer. Soc., *63*, 1737–52 (June, 1941).

2. Pepinsky, R., Robertson, J.M., and Speakman, J.C., editors, "Proceedings of Conference on Computing Methods and the Phase Problem in X-Ray Crystal Analysis Held at University of Glasgow in 1960," (Bergamon, New York, 1961), p. 45.

3. Additional information may be obtained through the contact person.

continued

4. Busing, W.R., Martin, K.O., and Levy, H.A., "A FORTRAN Crystallographic Least-Squares Program," ORNL Tech. Mem. No. TM305 (Oak Ridge National Laboratories, Oak Ridge, 1962). The modified version of this program is used for calculations that involve scattering data and crystallographic structures. Additional information can be obtained through the contact person.

5. Ibers, J.A., and Hamilton, W.C., "Dispersion Correction and Crystal Structure Refinement," Acta Cryst. *17*, (1964), 781–782.

SAMPLE INPUT

```
$ DATA
TEST STRUCTURE OF U MO C-2        ACTA. 17, 272 (1964)
   0     0     1     0      0  20.
 5.625 3.249  10.980                             .70926
   92.0000    91.5488    90.2096    88.2013    85.7514    83.0623    80.3408          1
   77.6645    75.0885    72.6012    70.2006    67.8789    65.6365    63.4953          2
   61.4569    59.5156    57.6656    55.9006    54.2128    52.5939    51.0467          3
   49.5686    48.1595    46.8144    45.5283    44.2965    43.1127    41.9782          4
   40.8910    39.8500    38.8523    37.8948    36.9744    36.0866    35.2322          5
   34.4101    33.6190    32.8575    32.1239    31.4164    30.7328    30.0727          6
   29.4350    28.8180    28.2207    27.6434    27.0857    26.5487    26.0298          7
   25.5274    25.0402      .0000      .0000      .0000      .0000      .0000          8
   42.0000    41.7204    40.8910    39.6552    38.1626    36.5453    34.9354          1
   33.3890    31.9567    30.6142    29.3460    28.1393    26.9869    25.8978          2
   24.8729    23.9170    23.0234    22.1838    21.3883    20.6265    19.9051          3
   19.2227    18.5804    17.9743    17.4005    16.8556    16.3349    15.8397          4
   15.3686    14.9208    14.4945    14.0876    13.6978    13.3208    12.9596          5
   12.6144    12.2868    11.9758    11.6782    11.3926    11.1143    10.8463          6
   10.5886    10.3421    10.1066     9.8801     9.6618     9.4491     9.2434          7
    9.0448     8.8534      .0000      .0000      .0000      .0000      .0000          8
    6.0000     5.9888     5.9401     5.8578     5.7461     5.6088     5.4502
    5.2741     5.0853     4.8883     4.6849     4.4785     4.2735     4.0731
    3.8775     3.6885     3.5109     3.3448     3.1883     3.0414     2.9044
    2.7771     2.6594     2.5511     2.4535     2.3648     2.2841     2.2107
    2.1432     2.0817     2.0258     1.9749     1.9279     1.8850     1.8459
    1.8103     1.7778     1.7481     1.7209     1.6952     1.6706     1.6476
    1.6262     1.6062     1.5873     1.5694     1.5522     1.5347     1.5174
    1.5006     1.4842     0.         0.         0.         0.         0.

         0          0          2   26.                         1    1
         0          0          4   278.                        1    0
         0          0          6  56.                          1    0
         0          0          8  181.                         1    0
         1          1          1  113.                         1    0
         1          1          2  333.                         1    0
         1          1          3  144.                         1    0
         1          2          1  49.                          1    0
         1          2          2  53.                          1    0
         1          2          3  154.                         1    0
         2          0          0  178.                         1    0
         2          0          1  210.                         1    0
         2          0          2  25.                          1    1
         2          1          0  124.                         1    0
         2          1          1  68.                          1    0
         2          1          2  80.                          1    0
         2          2          0  134.                         1    0
         2          2          1  170.                         1    0
         3          0          2  122.                         1    0
         3          0         12  183.                         1    0
         3          1         11  79.                          1    0
         3          3          1  151.                         1    0
         4          0          1  182.                         1    0
         4          1         10  155.                         1    0
         5          1          2  192.                         1    0

    1     4     1     0    13     2     0     0     0     0     1
    3     1     2     1     0
             1                  2                3
 .5         -1                 -2    0.5         3


1.0
U           1.0     1.0       .0819     .25       .1421     -15.  13.0
0.20
MO         2.0      1.0       .4191     .25       .8986     -1.7   0.9
0.39
C 1        3.0      1.0       .153      .25       .752
0.61
C 2        3.0      1.0       .751       .25      .005
0.00
100101100101100101100101l
```

```
TEST STRUCTURE OF U MO C-2      ACTA. 17, 272 (1964)
 A=   5.625, B=   3.249, C=  10.980, COSA= -.0000, COSB= -.0000, COSC= -.0000, LAMBDA=  .7093

HUGHES WEIGHTING,    FMIN=  20.0
ATOMIC SCATTERING FACTORS
            92.0    42.0     6.0     -.0      .0      .0      .0      .0      .0      .0
NUMBER OF SCALE FACTORS SPECIFIED IS      1
BASED ON F
ISOTROPIC AND ANISOTROPIC TEMP. FACTORS
NUMBER OF SCATTERING FACTOR TABLES IS  3
NUMBER OF ATOMS IN ASYMMETRIC UNIT IS  4
NUMBER OF ATOMS WITH DISPERSION CORRECTION IS   2
CENTROSYMMETRIC
NUMBER OF SYMMETRY CARDS IS  2
NUMBER OF SCALE FACTORS IS  1
SYMMETRY INFORMATION
    TRANSFORMED X          TRANSFORMED Y          TRANSFORMED Z

    -.000000 1-0           -.000000 2-0           -.000000 3-0
     .500000-1-0           -.000000-2-0            .500000 3-0
NUMBER OF PARAMETERS READ IS  25
NUMBER OF PARAMETERS TO BE REFINED IS   13
INPUT DATA
SCALE   P(I)  KI(I)
  1   1.0000   1
ATOM     F         O          X          Y          Z        B(B11)     B22       B33       B12       B13       B23

  1     1.00 0   1.0000 0   .0819 1   .2500 0   .1421 1   .2000 1
  2     2.00 0   1.0000 0   .4191 1   .2500 0   .8986 1   .3900 1
  3     3.00 0   1.0000 0   .1530 1   .2500 0   .7520 1   .6100 1
  4     3.00 0   1.0000 0   .7510 1   .2500 0   .0050 1   .0000 1
```

continued

```
TEST STRUCTURE OF U MO C-2      ACTA. 17, 272 (1964)

CALCULATED F BASED ON PARAMETERS BEFORE CYCLE 1

   H   K   L    F(OBS)     F(CALC)        A           B           SINE     SIG(O)  (O-C)/SIG(O)   IQ   ID

   0   0   2    26.0000    20.5575   -17.9495    -10.0212         .065    80.0000       .0680      1    1
   0   0   4   278.0000   299.9247  -295.7436    -49.9053         .129   278.0000     -.0789      1    0
                        .                      .           .
                        .                      .           .
                        .                      .           .
   3   3   1   151.0000   159.2828   155.7473     33.3733         .380   151.0000     -.0549      1    0
   4   0   1   182.0000   194.7788  -191.2908    -36.6962         .254   182.0000     -.0702      1    0
   4   1  10   155.0000   152.6271   147.1829     40.4007         .424   155.0000      .0153      1    0
   5   1   2   192.0000   192.8319   187.8644     43.4871         .340   192.0000     -.0043      1    0

AGREEMENT FACTORS BASED ON PARAMETERS BEFORE CYCLE 1

SUM(W*(O-C)**2) IS   .6939E-01

SQRTF(SUM(W*(O-C)**2)/(NO-NV)) IS      .0760

                                            NUMERATOR   DENOMINATOR     R

R FACTOR INCLUDING UNOBS                     154.164      3440.000     .045

R FACTOR OMITTING UNOBS                      146.349      3389.000     .043

WEIGHTED R FACTOR INCLUDING UNOBS               .263         4.605     .057

WEIGHTED R FACTOR OMITTING UNOBS                .253         4.583     .055

TEST STRUCTURE OF U MO C-2      ACTA. 17, 272 (1964)

PARAMETERS AFTER LEAST SQUARES CYCLE 1

 PARAMETER          OLD        CHANGE         NEW         ERROR

SCALE FACTOR    1.0000000    .0037115    1.0037115    .0283736

U

FORM FACTOR     1.0000000                1.0000000

MULTIPLIER      1.0000000                1.0000000

X                .0819000   -.0012466     .0806533    .0016965

Y                .2500000                 .2500000

Z                .1421000    .0004068     .1425068    .0008369

ATOMIC B         .2000000   -.0092766     .1907234    .1886939
             .                          .           .
             .                          .           .
             .                          .           .
C 2

FORM FACTOR     3.0000000                3.0000000

MULTIPLIER      1.0000000                1.0000000

X                .7510000   -.0301996     .7208003    .0225634

Y                .2500000                 .2500000

Z                .0050000   -.0061485    -.0011485    .0173400


ATOMIC B         .0000000   1.6818284    1.6818284   1.9679505

ESTIMATED AGREEMENT FACTORS BASED ON PARAMETERS AFTER CYCLE 1

SUM(W*(O-C)**2) IS   .3529E-01

SQRTF(SUM(W*(O-C)**2)/(NO-NV)) IS      .0542
```

continued

```
TEST STRUCTURE OF U MO C-2       ACTA. 17, 272 (1964)

CALCULATED F BASED ON PARAMETERS BEFORE CYCLE 2

   H    K    L     F(OBS)     F(CALC)         A           B             SINE      SIG(O)  (O-C)/SIG(O)    IQ    ID

   0    0    2    26.0965    20.0279    -17.2174    -10.2313           .065     80.0000       .0756       1     1
   0    0    4   279.0318   298.0844   -293.9271    -49.6104           .129    278.0000      -.0683       1     0
   0    0    6    56.2078    57.9914     50.4084     28.6706           .194     80.0000      -.0222       1     0
   0    0    8   181.6718   176.9955    173.8244     33.3538           .258    181.0000       .0257       1     0
   1    1    1   113.4194   111.5944   -110.2726    -17.1248           .130    113.0000       .0161       1     0
   1    1    2   334.2359   314.6329   -311.1165    -46.9083           .142    333.0000       .0587       1     0
   1    1    3   144.5345   145.2431    143.4101     23.0027           .159    144.0000      -.0049       1     0
   1    2    1    49.1819    48.7262    -41.5077    -25.5215           .230     80.0000       .0057       1     0
   1    2    2    53.1967    52.8388     47.8116     22.4942           .236     80.0000       .0045       1     0
   1    2    3   154.5716   137.8471    132.2144     39.0024           .247    154.0000       .1082       1     0
   2    0    0   178.6607   176.7828    174.3810     29.0419           .126    178.0000       .0105       1     0
   2    0    1   210.7794   217.1951   -214.1845    -36.0378           .130    210.0000      -.0304       1     0
   2    0    2    25.0928    25.2177    -24.6249     -5.4356           .142     80.0000      -.0016       1     1
   2    1    0   124.4602   129.1477   -122.5902    -40.6296           .167    124.0000      -.0377       1     0
   2    1         68.2524    70.5149    -67.5469    -20.2428           .170     80.0000      -.0282       1     0
   2    1    2    80.2969    83.3624     82.7058     10.4424           .179     80.0000      -.0382       1     0
   2    2    0   134.4973   133.5550   -130.4895    -28.4506           .252    134.0000       .0070       1     0
   2    2    1   170.6310   164.6388    160.8063     35.3167           .254    170.0000       .0351       1     0
   3    0    2   122.4528   120.4900   -111.0855    -46.6674           .200    122.0000       .0160       1     0
   3    0   12   183.6792   179.2926    172.2991     49.5869           .431    183.0000       .0239       1     0
   3    1   11    79.2932    81.3062     69.6187     41.9991           .417     80.0000      -.0251       1     0
   3    3    1   151.5604   150.1268    146.4191     33.1587           .380    151.0000       .0095       1     0
   4    0    1   182.6755   188.3604   -184.6502    -37.2011           .254    182.0000      -.0311       1     0
   4    1   10   155.5753   156.7659    151.2343     41.2762           .424    155.0000      -.0077       1     0
   5    1    2   192.7126   189.7096    184.8885     42.4972           .340    192.0000       .0156       1     0

AGREEMENT FACTORS BASED ON PARAMETERS BEFORE CYCLE 2

SUM(W*(O-C)**2) IS    .3582E-01

SQRTF(SUM(W*(O-C)**2)/(NO-NV)) IS       .0546

                                                  NUMERATOR   DENOMINATOR       R

R FACTOR INCLUDING UNOBS                            116.299      3452.768     .034

R FACTOR OMITTING UNOBS                             110.105      3401.578     .032

WEIGHTED R FACTOR INCLUDING UNOBS                      .189         4.605     .041

WEIGHTED R FACTOR OMITTING UNOBS                       .174         4.583     .038
```

continued

```
TEST STRUCTURE OF U MO C-2      ACTA. 17, 272 (1964)

CORRELATION MATRIX

  1   1.0000    .0424   -.1581    .7396   -.1126   -.0477    .2227    .0030   -.1603    .2436   -.0826    .0598    .2489

  2   1.0000    .0655    .0513    .1258   -.0040   -.1101    .1467    .0723    .0797    .0907    .0042    .1213
  3   1.0000   -.1651   -.0552    .0278   -.1087    .0916   -.1553   -.4591   -.2431    .3275    .2424
  4   1.0000   -.0635   -.3191    .0740   -.0222    .0106    .1986   -.0746   -.0870    .4436
  5   1.0000   -.0040   -.0592    .3686    .3758    .0534    .3955   -.0963   -.2244
  6   1.0000    .3857   -.0936    .1643    .0846   -.0805   -.1663   -.1475
  7   1.0000   -.1715   -.0166    .1372   -.1361   -.2988   -.2061
  8   1.0000    .0396    .1430    .2138    .2034   -.0466
  9   1.0000   -.0286    .1441   -.4963   -.0912
 10   1.0000    .3886   -.0739   -.0410
 11   1.0000   -.0326   -.1519
 12   1.0000   -.0781
 13   1.0000
```

continued

```
DISAGREEMENT INDEX FOR VARIOUS GROUPS


SINE THETA UP TO        0.1     0.2     0.3     0.4     0.5     0.6     0.7     0.8     0.9     1.0

NO. REFLEXION             1      12       7       2       3       0       C       0       0       0
R FACTOR               .233    .036    .038    .013    .018    .000    .000    .000    .000    .000

                        HKL     HKO     OKL     HOL     HOO     OKO     OOL     ALL

NO. REFLEXION            13       2       0       5       1       0       4      25
R FACTOR               .034    .022    .000    .026    .011    .000    .058    .034

                        EEE     EEO     EOE     OEE     EOO     OEO     OOE     OOO

NO. REFLEXION             7       3       3       3       1       2       2       4
R FACTOR               .039    .032    .025    .019    .033    .084    .043    .012

OBS. ONLY    SUM FOBS    SUM FCAL    SCALE,  UNOBS. ONLY    SUM FOBS    SUM FCAL    TOTAL SCALE
              3401.58     3385.20     .995                    51.19       45.25          .994


F VALUE UP TO           10.     20.     30.     40.     50.     60.     70.     80.     90.    100. OVER 100

NO. REFLEXION             0       0       0       0       1       2       1       1       1       0      17
AVER. W*DELTSQ         .000    .000    .000    .000    .000    .000    .001    .001    .001    .000    .002

NO. REFLEXION             0       0       2       0       0       0       0       0       0       0       0
AVER. W*DELTSQ         .000    .000    .003    .000    .000    .000    .000    .000    .000    .000    .000


SINE THETA UP TO        0.1     0.2     0.3     0.4     0.5     0.6     0.7     0.8     0.9     1.0

NO. REFLEXION             0      11       7       2       3       0       0       0       0       0
AVER. W*DELTSQ         .000    .001    .002    .000    .000    .000    .000    .000    .000    .000
NO. REFLEXION             1       1       0       0       0       0       0       0       0       0
AVER. W*DELTSQ         .006    .000    .000    .000    .000    .000    .000    .000    .000    .000
```

COST ESTIMATE

Unfortunately, exact figures are not available for central-processor time. The program listed on the Sample Input used approximately 10.8 minutes of central-processor time. At the current rate of $250/hour at the University of Pittsburgh, the chargeable computer time was approximately $45.00.

Cost to user = computer time + network overhead

= $45.00 (estimated) + network overhead

CONTENTS—ORFLS-PX

pages

pages	
1– 3	Identification & Abstract
5–22	User Instructions
23–28	I/O
29	Cost—Contents

DESCRIPTIVE TITLE	Factor Analysis by Direct "Oblimin" Method
CALLING NAME	OBLIMIN
INSTALLATION NAME	Educational Testing Service (ETS) Office of Data Analysis Research
AUTHOR(S) AND AFFILIATION(S)	Procedure due to: R. Jennrich, UCLA Program due to: R. Jennrich and H. Harman, ETS Adaptation due to: J. Barone, ETS
LANGUAGE	FORTRAN IV
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Office of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

Factor analysis provides that the final solution be in terms of either uncorrelated factors or correlated factors. Beginning in the mid-1940's, following the leadership of Thurstone, there was a trend toward the acceptance of oblique factors. This trend has continued to the present day but, unfortunately, efficient objective means for getting oblique "simple structure" solutions have not generally been available even with modern computers. In 1958, John B. Carroll introduced a whole class of methods for oblique transformation to simple structure. These have come to be known by the term "oblimin" (Ref. 1, pp. 324–326), since they involve *oblique* factors and the *minimization* of a function. The oblimin criterion, which is to be minimized, is given in normalized form by,

1. $$B = \sum_{p<q=1}^{m} \left[n \sum_{j=1}^{n} (v_{jp}^2/h_j^2)(v_{jq}^2/h_j^2) - \gamma \sum_{j=1}^{n} v_{jp}^2/h_j^2 \sum_{j=1}^{n} v_{jq}^2/h_j^2 \right]$$

continued

The v's in these expressions are the elements of the reference-factor structure matrix, i.e., the correlations between the original variables and the reference factors. Actually, what is desired is to have the primary factor pattern exhibit the principles of simple structure, i.e., large values and near-zero values. In the Thurstone school the "reference structure V" is sought that exhibits the simple structure principles, and then the primary factor pattern is obtained by multiplying the matrix V by a diagonal matrix.

The foregoing indirect, and somewhat awkward, procedure has recently been replaced by a direct approach (Ref. 2). Instead of working with the reference factors that are biorthogonal to the primary factors, Jennrich and Sampson set up a criterion for the direct determination of the primary factors that exhibit the simple structure principles. That criterion may be put in the form,

$$2. \qquad F(A) = \sum_{p<q=1}^{m} \left[\sum_{j=1}^{n} a_{jp}^{2} a_{jq}^{2} - \frac{\delta}{n} \sum_{j=1}^{n} a_{jp}^{2} \sum_{j=1}^{n} a_{jq}^{2} \right] ,$$

where A is the matrix of primary factor coefficients. Of course, the loadings may be normalized by rows just as in equation 1. An important difference is that the γ in the indirect method ranges between zero and one, while the δ in equation 2 should be zero or negative.

The object of OBLIMIN is to minimize equation 2. The criterion employed is,

$$\frac{F_{i-1} - F_{i}}{F_{0}} \leqq \varepsilon \quad ,$$

where i is the iteration number. The output is an oblique factor solution satisfying the principles of simple structure, more or less. When δ is equal to zero, the factors are most oblique. For negative values of δ, the factors become less oblique as δ gets smaller. The solution consists of the factor pattern, the correlations among the factors, and the factor structure. At the present time, the program is limited to n = 100 variables and m = 15 factors.

REFERENCES

Harman, H.H., *Modern Factor Analysis*, Second edition, revised, (University of Chicago Press, Chicago, Ill., 1967)

Jennrich, R.I., & Sampson, P.F., "Rotation for Simple Loadings," *Psychometrika*, 1966, *31*, 313–323.

USER INSTRUCTIONS

System Control Cards

JOB Card will be prepared by ETS personnel.

System Card 1

Columns	*Contents*
1–37	//STEPNAME EXEC GITNGO,NAME=OBLIMIN

System Card 2

Columns	*Contents*
1–19	//GO.SYSIN DD *

System Card 3

Columns	*Contents*
1– 2	//

Problem Card

Parameter	*Columns*	*Format*	*Description*
N	3– 5	I3	Number of variables (≤ 100)
M	6–10	I5	Number of factors (≤ 15)
DEL	11–15	F5.1	Delta (except for experimental purposes, $\delta \leq 0$; if several values are used, smallest goes in this field)
EPS	16–25	F10.6	Convergence criterion (usually = .00001)
MAXI	26–30	I5	Maximum number of iterations (usually 50)
IF	31–35	I5	Original factor pattern option: 1 = orthogonal (usual case, requires no additional input) 0 = oblique (correlations among the original factors must be input in addition to the pattern)

continued

Parameter	*Columns*	*Format*	*Description*
K	36–40	I5	Kaiser normalization option: 1 = used (usual case) 0 = not used
DINC	41–45	F5.1	Delta increment (must be a positive number)
DEND	46–50	F5.1	Largest delta
NOSTOP	51–55	I5	1 = additional problem(s) to follow 0 = last problem

NOTE: The last three fields may be blank if only one value of delta and one problem are run.

Format Card

The format specification starts in Col. 2 and may not extend beyond Col. 72. The word FORMAT is *not* used. The standard FORTRAN variable format is followed.

Data Cards

Keypunch original factor pattern by rows according to its format, starting each variable on a new card.

Factor Correlations Cards

When the original factor pattern is oblique, a "0" must be put in Col. 35 of the Problem Card, and the following cards must be inserted at this point,

(a) FORMAT Cards (Cols. 2–72)

(b) Correlations among factors, starting a new card for each factor

Title Card

Name of problem in Cols. 2–80.

Order of Cards

JOB Card
System Card 1
System Card 2
Problem Card
Format Card

continued

Data Cards
Factor Correlations Cards
Title Card
/*
System Card 3

Description of Output

See Functional Abstract for a description of the output of OBLIMIN.

SAMPLE INPUT

```
//            JOB  (....
//STEPNAME   EXEC  GITNGO,NAME=OBLIMIN
//GO.SYSIN   DD     *
       8     2  0.0  0.000010    50     1     1 -0.5 -2.0     0
 (2F10.5)
   0.85600  -0.32400
   0.84800  -0.41200
   0.80800  -0.40900
   0.83100  -0.34200
   0.75000   0.57100
   0.63100   0.49200
   0.56900   0.51000
   0.60700   0.35100
/*
//
```

SAMPLE OUTPUT

```
DELTA =        0.00

ROTATION FOR DIRECT OBLIMIN LOADINGS

ITERATION          DIROBL
                 CRITERION
     0            2.905183
     1            2.683994
     2            0.731741
     3            0.072721
     4            0.072715

AFTER ROTATION WITH KAISER NORMALIZATION

FACTOR PATTERN

     1    0.88272    0.06520
     2    0.95632   -0.02946
     3    0.92599   -0.04505
     4    0.88163    0.03492
     5    0.00471    0.94040
     6   -0.00647    0.80317
     7   -0.06549    0.79280
     8    0.10374    0.64627

FACTOR CORRELATIONS

     1    1.00000    0.47149
     2    0.47149    1.00000

FACTOR STRUCTURE

     1    0.91346    0.48139
     2    0.94243    0.42143
     3    0.90475    0.39154
     4    0.89810    0.45060
     5    0.44809    0.94262
     6    0.37221    0.80012
     7    0.30831    0.76192
     8    0.40845    0.69518
```

continued

```
DELTA =      -0.50

ROTATION FOR DIRECT OBLIMIN LOADINGS

                  .
                  .
                  .


DELTA =      -1.00

ROTATION FOR DIRECT OBLIMIN LOADINGS

                  .
                  .
                  .


DELTA =      -1.50

ROTATION FOR DIRECT OBLIMIN LOADINGS

                  .
                  .
                  .


DELTA =      -2.00

ROTATION FOR DIRECT OBLIMIN LOADINGS

ITERATION   DIROBL
         CRITERION
     0    9.111930     9       7.357594
     1    9.050656    10       7.303166
     2    8.985585    11       7.282603
     3    8.874997    12       7.275172
     4    8.695285    13       7.272531
     5    8.426510    14       7.271598
     6    8.080988    15       7.271269
     7    7.735189    16       7.271153
     8    7.487005    17       7.271112
AFTER ROTATION WITH KAISER NORMALIZATION
```

continued

FACTOR PATTERN

1	0.86186	0.14201
2	0.92372	0.05725
3	0.89278	0.03949
4	0.85778	0.11269
5	0.09845	0.90789
6	0.07392	0.77446
7	0.01569	0.75917
8	0.16504	0.63294

FACTOR CORRELATIONS

1	1.00000	0.30538
2	0.30538	1.00000

FACTOR STRUCTURE

1	0.90522	0.40520
2	0.94121	0.33934
3	0.90484	0.31213
4	0.89219	0.37464
5	0.37570	0.93795
6	0.31043	0.79704
7	0.24753	0.76396
8	0.35833	0.68334

COST ESTIMATE

Exact figures for the program listed on the Sample Input are not available. As an approximate guide to the user, an OBLIMIN run whose parameters lie within the specified range will cost less than $20.00 with a typical cost of less than $10.00.

Approximate cost to user = computer time + postage and handling + network overhead
= $10.00 + $5.00 + network overhead
= $15.00 + network overhead

CONTENTS—OBLIMIN

pages

1– 2 Identification & Abstract
3– 5 User Instructions
7–10 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Unrestricted Maximum Likelihood Factor Analysis
CALLING NAME	UMLFA
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service
AUTHOR(S) AND AFFILIATION(S)	K. G. Jöreskog Educational Testing Service G. Gruvaeus Educational Testing Service
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Office of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, New Jersey 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

UMLFA performs a factor analysis of a given correlation matrix. "The factor loadings and the unique variances are estimated by Lawley's method of maximum likelihood. The computational procedure...makes use of the method of Fletcher and Powell for numerical minimization of a function. Any number of factors can be extracted, and each factor is rotated, using Kaiser's varimax method. The goodness of fit of the maximum likelihood solution is tested by Lawley's chi-square test based on the likelihood ratio technique."[1]

The program is able to handle up to 75 variables and 30 factors. However, the input correlation matrix must be positive definite.

REFERENCES

1. Jöreskog, K.G., "A Computer Program for Unrestricted Maximum Likelihood Factor Analysis," Educ. Testing Service Res. Mem. RM-66-20 (1966).

USER INSTRUCTIONS

Specification of Input

360 O.S./JCL cards (job control)

```
Col. 1         Col. 16

//          JOB
//JOBLIB    DD  DSN=OCS.SAPGM,DISP=SHR
//GO      EXEC  PGM=UMLFA,TIME=2,REGION=150K,ACCT=xxxxx
//FT06F001 DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//FT07F001 DD   SYSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520)
//FTXXF001 DD   SPACE=(CYL,(1,1)),DISP=NEW,UNIT=2314
                    [Note:  Replace XX in above card with
                            number punched in Cols. 29-30
                            of parameter card no. 2.]
//FT05F001 DD   *
         (UMLFA control cards and data go here)
/*
//
```

Heading Card

Any information may be punched in Cols. 1–72.

Parameter Card

Column	*Value or Variable*	*Description*	*(Recommended Value)*
1– 5	P	Number of variables	
6–10	K	Lower bound for number of factors	(≥ 1)
11–15	K	Upper bound for number of factors	(≤ 30)
16–20	NP	Sample size	
21–25	MAXITER	Maximum number of iterations	(safely, 3P)
26–30		Scratch tape number	
31–35	CRV	Convergence criterion	(usually 1.0)
41	1/0	Print/don't print original correlation matrix and partial correlation matrices with some variables eliminated	
42	1/0	Print/don't print technical output related to the iterative process	(usually 0)

continued

43	1/0	Print/don't print intermediate results when some variables are eliminated, and the reproduced correlations and residuals for the final solution
44	1/0	Punch/don't punch unrotated factor matrix (0, unless it is desired to use the result for input to another rotation program)
45	1/0	Suppress/don't suppress boundary elimination
46	1/0	Nonstandard/standard parameters
47	1/0	Read/don't read starting vector of uniquenesses
48	1/0	Nonstandard/standard output
49	1/0	Calculates/does not calculate to more than 3-place accuracy by Ortega's method
50–80		Must be blank

Columns 45–49 are intended for experimentation and ordinarily would be left blank.

Nonstandard Parameter Cards (optional)

If Col. 46 in the parameter cards is 1, two nonstandard parameter cards must be inserted at this point. The user should refer to the write-up for the definition of these parameters.

Data-Format Card

The standard FORTRAN variable format is followed; i.e., a card of the form (ηFω.d) must be punched.

Correlation Cards

The correlation cards punched according to the format specified in nonstandard parameter cards must be prepared in row order, each row beginning on a new card. Only the lower half of the correlation matrix *excluding* the main diagonal is punched; i.e., there will be P - 1 rows of input.

Starting Configuration (optional)

If Col. 47 in the parameter card is 1, an arbitrary starting configuration may be entered. This will consist of

(a) a format card

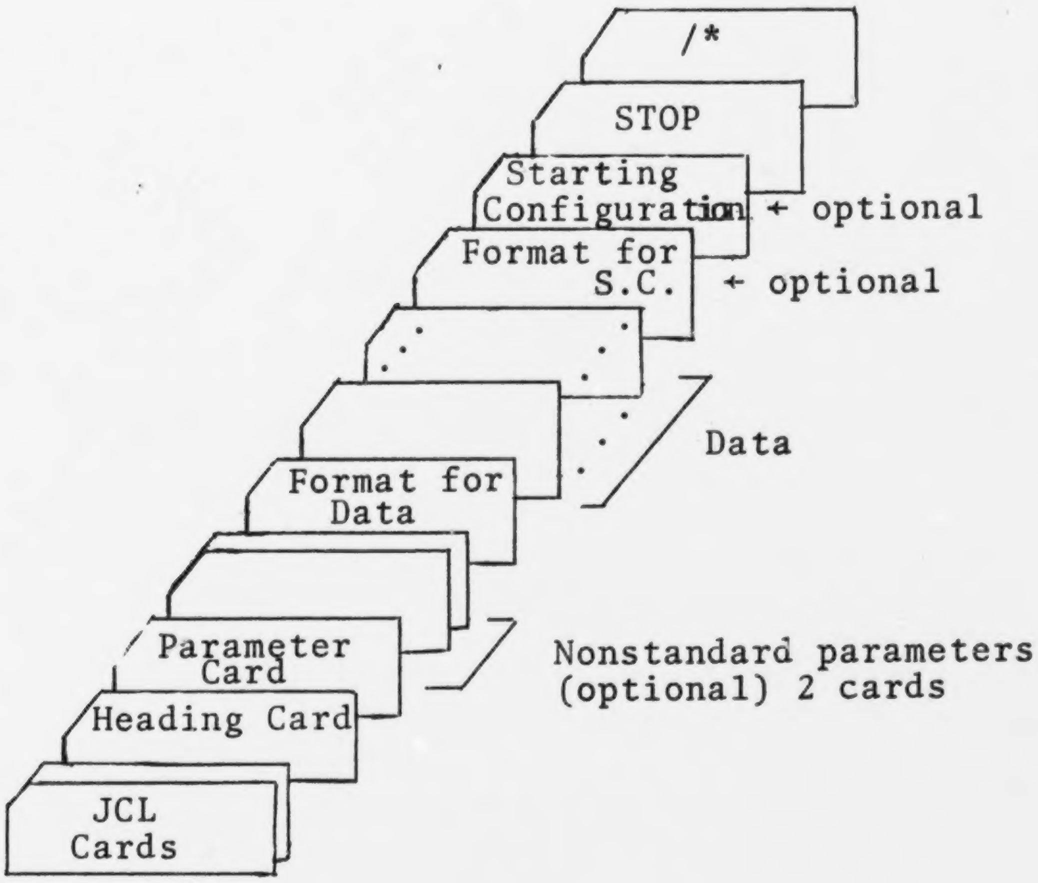
(b) the starting uniquenesses for the variables punched according to the format (a) above.

continued

Stop Card

At the end of the last case, a card punched STOP in Cols. 1–4 must be placed.

The input deck should be in the following order.


/*
STOP
Starting Configuration
← optional
Format for S.C.
← optional
Data
Format for Data
Nonstandard parameters (optional) 2 cards
Parameter Card
Heading Card
JCL Cards

Description of Output

The output is dependent upon the options selected in Cols. 41–44 of the parameter card. For purposes of illustration the sample input did not contain any punches after Col. 40, so that a minimum output resulted. The following items are printed:

(1) input parameters

(2) maximum likelihood solution (labeled "unrotated factor matrix")

(3) uniqueness for each variable

(4) varimax solution

continued

(5) a test of goodness of fit for the specified number of factors.

Note: With items (2), (3), and (4) there appear a string of numbers for diagnostic purposes. *These should be ignored!*

REFERENCES

Jöreskog, K.G., "A Computer Program for Unrestricted Maximum Likelihood Factor Analysis," Educational Testing Service Res. Mem. RM-66-20 (1966).

SAMPLE INPUT

A listing of the input data cards and the related output are presented below. Although the data used have the entire matrix punched, it would not have been necessary. Data punched through the diagonal value for each row would have been sufficient.

```
EIGHT VARIABLES--ALL 41 ON BLANK
    8    3    3  305   40    4 1.0
 (3X,8F8.5)
 21 0.84600 1.00000 0.88100 0.82600 0.37600 0.32600 0.27700 0.41500
 31 0.80500 0.88100 1.00000 0.80100 0.38000 0.31900 0.23700 0.34500
 41 0.85900 0.82600 0.80100 1.00000 0.43600 0.32900 0.32700 0.36500
 51 0.47300 0.37600 0.38000 0.43600 1.00000 0.76200 0.73000 0.62900
 61 0.39800 0.32600 0.31900 0.32900 0.76200 1.00000 0.58300 0.57700
 71 0.30100 0.27700 0.23700 0.32700 0.73000 0.58300 1.00000 0.53900
 81 0.38200 0.41500 0.34500 0.36500 0.62900 0.57700 0.53900 1.00000
STOP
```

SAMPLE OUTPUT

```
UNRESTRICTED MAXIMUM LIKELIHOOD FACTOR ANALYSIS

EIGHT VARIABLES--ALL 41 ON BLANK

   5/29/68

PARAMETERS


P=    8

K1=    3

K2=    3

N=  305

MAXITER=   40

CRV= 1.000

MAXIMUM LIKELIHOOD SOLUTION FOR   3 FACTORS

UNROTATED FACTOR MATRIX

   0    8    3  305    5    9
0.846   -0.189      0.348
1.000    0.000      0.000
0.881   -0.055      0.164
0.826   -0.160      0.368
0.376   -0.876     -0.031
0.326   -0.725     -0.093
0.277   -0.704     -0.130
0.415   -0.543     -0.204
```

continued

```
 UNIQUE VARIANCES

    0     8  305     6     9
0.127     0.000     0.194     0.156     0.090     0.359     0.411     0.491

 VARIMAX-ROTATED FACTOR MATRIX

    0     8     3  305     5     9
0.883     0.275    -0.129
0.936     0.204     0.287
0.872     0.196     0.088
0.875     0.239    -0.145
0.234     0.919    -0.105
0.185     0.779    -0.021
0.129     0.757     0.004
0.254     0.648     0.157


TEST OF GOODNESS OF FIT FOR   3 FACTORS


CHISQUARE WITH     8 DEGREES OF FREEDOM IS         22.5993

PROBABILITY LEVEL IS 0.00
```

COST ESTIMATE

For the job listed on the Sample Input, the total running time on the IBM 360/65 amounted to 3.80 seconds. At the current rate at Educational Testing Service (a rather complicated function of the ratio of CPU time to actual wall-clock running time), the total computer charge for the run amounted to $10.40. (The wall-clock/CPU-clock ratio was approximately 25:1.)

Charge to user = computer time + postage and handling + network overhead

= $10.40 + $5.00 + network overhead = $15.40

CONTENTS—UMLFA

pages

1 Identification & Abstract
3–6 User Instructions
7–9 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Rao Constellation and Distance Analysis
CALLING NAME	DISCRIM2
INSTALLATION NAME	Michigan State University Computer Institute for Social Science Research
AUTHOR(S) AND AFFILIATION(S)	Stuart Thomas Computer Institute for Social Science Research Michigan State University
LANGUAGE	FORTRAN
COMPUTER	CDC 3600
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Dr. Anders Johanson, Programming Supervisor, Applications Programming, Computer Laboratory, Computer Center, Michigan State University, East Lansing, Mich. 48823 Tel.: (517) 355–4684

FUNCTIONAL ABSTRACT

This program implements the method titled "Constellation and Distance Analysis" when first published by C.R. Rao[1] and later called "Multiple Discriminant Analysis" when presented independently by Bryan[2] and by Lubin. The method operates on a set of variates measured on individuals in several groups. It determines linear combinations of the variates, called discriminant functions, which maximize the ratio of between-group variability to pooled, within-group variability, producing the output listed below. The user can have the program handle data input or write his own subroutine to read data and perform preliminary data manipulations.

Output

1. Job Description

   User comments
   Number of groups and variables
   Options selected

continued

2. Group Summary Statistics

   Identification
   Number observations specified on group card
   Mode of data input and input unit
   Format statement (if data read by program rather than user)
   First observation in the group (raw and transformed)
   Variable means and variances
   Variable intercorrelation matrix (optional)

3. Overall Data Summary

   Overall means
   Overall variances
   Overall correlation matrix

4. Discriminant Function Information

   Discriminant criterion
   Percent trace accounted for by criterion
   Rao's chi-square and degrees of freedom for the function
   Function weights for raw data
   Function weights for data adjusted to unit variances (optional)
   Mean discriminant score for each group

5. Overall Discriminant Statistics

   Group centroids in discriminant space (optional)
   Intercentroid distance matrix (optional)
   Back solution of discriminant equation (optional)

6. Discriminant Scores (optional)

Capacity

Number of variables must not exceed 30
Number of groups must not exceed 50
Number of observations in any given group must not exceed 9,999
Number of scores (if requested) must not exceed 10 per respondent

REFERENCES

1. Rao, C.R., *Advanced Statistical Methods in Brometric Research*, (John Wiley & Sons, Inc., New York, 1952) Ch. 9.

2. Bryan, J.G., "The Generalized Discriminant Function: Mathematical Foundation and Computational Routine," Harvard Educational Review, *21*, 2, (1951), 90–95.

USER INSTRUCTIONS

DISCRIM is called from the CISSR library tape by the name DISCRIM2. The routine is available in relocatable object deck form; however, unless extensive recoding of data is required, use of the library tape is recommended. The user may supply his own recoding subroutine for use with the object deck form.

Complete instructions on the options available and the use of the program are found in the references cited below.

REFERENCES

"CISSR Technical Report No. 1," (The Mich. State Univ., East Lansing, Mich.).

Thomas, S., "DISCRIM2, CISSR Technical Report No. 33, (The Mich. State Univ., East Lansing, Mich.). Available from the EIN Office at the cost of reproduction and mailing.

SAMPLE INPUT

```
'JOB,015525,THOMAS,3.0,APHL
'EQUIP,27=(CISSR),MT(193),RO
'LOADMAIN,27,3.0,3000
$ROUTINE,DISCRIM2
$DISCRIM TEST DATA
SIZE      3    5    0    1    1    1    1    1    1
SCORES    2    PU
GROUP     1    5    0    1    1
(5F5.0)
    1     2    3    4    5
    2     4    5    7    9
    3     6    9   12   16
    4     5    7    9    1
    5     9    7    8    6
GROUP     2    5    0    1    1
(5F5.0)
   11    13   15   17   19
   12    14   15   16
   13    13   13   13   13
   14    15   15   19   21
   15    16   17   18   19
GROUP     3    4    0    1    1
(5F5.0)
   21    40   50   60   70
   22    43   49   65   71
   23    45   48   64   72
   24    47   47   66   73
ENDATA
```

SAMPLE OUTPUT

```
3                                         MULTIPLE DISCRIMINANT ANALYSIS
         IMPLEMENTED BY COMPUTER INSTITUTE FOR SOCIAL SCIENCE RESEARCH
                                          MICHIGAN STATE UNIVERSITY

                DISCRIM TEST DATA
SIZE CARD SPECIFIES  5 VARIABLES, 3 GROUPS
SCORES CARD REQUESTS  2 SCORES PER RESPONDENT OUTPUT TO 00



                 DATA SUMMARY FOR GROUP     1
SIZE CARD SPECIFIES         5 OBSERVATIONS
FORMAT FOR GROUP        1 ** (5F5,0)
DATA INPUT BY PROGRAM  FROM LOGICAL UNIT 60
FIRST RAW OBSERVATION FOR GROUP     1

 VARIABLE       1           2           3           4           5
 VALUE      1,0000+000  2,0000+000  3,0000+000  4,0000+000  5,0000+000
FIRST TRANSFORMED OBSERVATION FOR GROUP     1

 VARIABLE       1           2           3           4           5
 VALUE      1,0000+000  2,0000+000  3,0000+000  4,0000+000  5,0000+000
COMPUTATIONS BASED ON       5 OBSERVATIONS AFTER DROPPING        0
VARIABLE MEANS AND VARIANCES IN GROUP     1
 VARIABLE       1           2           3           4           5
 MEAN       3,0000+000  5,2000+000  6,2000+000  8,0000+000  7,4000+000
 VARIANCE   2,5000+000  6,7000+000  5,2000+000  8,5000+000  3,1300+001

CORRELATION MATRIX FOR GROUP     1



            1           2           3           4           5

    1  1,0000+000  9,1627-001  6,9338-001  5,4233-001 -1,6957-001
    2  9,1627-001  1,0000+000  7,1156-001  5,6317-001  1,4847-001
    3  6,9338-001  7,1156-001  1,0000+000  9,7769-001  4,6246-001
    4  5,4233-001  5,6317-001  9,7769-001  1,0000+000  5,5177-001
    5 -1,6957-001  1,4847-001  4,6246-001  5,5177-001  1,0000+000
```

continued

```
                       DATA SUMMARY FOR GROUP      2
SIZE CARD SPECIFIES         5 OBSERVATIONS
* FORMAT FOR GROUP      2 *-  (5F5,0)
DATA INPUT BY PROGRAM  FROM LOGICAL UNIT 60
FIRST RAW OBSERVATION FOR GROUP      2

 VARIABLE       1           2           3           4           5

 VALUE      1,1000+001  1,3000+001  1.5000+001  1.7000+001  1.9000+001
FIRST TRANSFORMED OBSERVATION FOR GROUP      2

 VARIABLE       1           2           3           4           5

 VALUE      1,1000+001  1,3000+001  1.5000+001  1.7000+001  1.9000+001
COMPUTATIONS BASED ON        5 OBSERVATIONS AFTER DROPPING           0
VARIABLE MEANS AND VARIANCES IN GROUP      2

 VARIABLE       1           2           3           4           5
 MEAN       1,3000+001  1,4200+001  1.5000+001  1.6600+001  1.4400+001
 VARIANCE   2,5000+000  1,7000+000  2,0000+000  5.3000+000  7.3800+001

CORRELATION MATRIX FOR GROUP      2



           1           2           3           4           5

   1  1,0000+000  8,4887-001  4,4721-001  3,4340-001  3.8651-001

   2  8,4887-001  1,0000+000  8,1349-001  6,9961-001  2,8123-001

   3  4,4721-001  8,1349-001  1,0000+000  7,6787-001  2,4693-001

   4  3,4340-001  6,9961-001  7,6787-001  1,0000+000  4,7782-001

   5  3,8651-001  2,8123-001  2,4693-001  4,7782-001  1,0000+000



                       DATA SUMMARY FOR GROUP      3
SIZE CARD SPECIFIES         4 OBSERVATIONS
FORMAT FOR GROUP      3 *-  (5F5,0)
DATA INPUT BY PROGRAM  FROM LOGICAL UNIT 60
FIRST RAW OBSERVATION FOR GROUP      3

 VARIABLE       1           2           3           4           5

 VALUE      2,1000+001  4,0000+001  5.0000+001  6,0000+001  7.0000+001
FIRST TRANSFORMED OBSERVATION FOR GROUP      3

 VARIABLE       1           2           3           4           5

 VALUE      2,1000+001  4,0000+001  5.0000+001  6,0000+001  7.0000+001
COMPUTATIONS BASED ON        4 OBSERVATIONS AFTER DROPPING           0
```

continued

```
VARIABLE MEANS AND VARIANCES IN GROUP     3

 VARIABLE        1            2            3            4            5
 MEAN       2,2500+001   4,3750+001   4,8500+001   6,3750+001   7,1500+001
 VARIANCE   1,6667+000   8,9167+000   1,6667+000   6,9167+000   1,6667+000

CORRELATION MATRIX FOR GROUP     3



            1            2            3            4            5

   1  1,0000+000   9,9438-001  -1,0000+000   8,3450-001   1,0000+000

   2  9,9438-001   1,0000+000  -9,9438-001   8,8074-001   9,9438-001

   3 -1,0000+000  -9,9438-001   1,0000+000  -8,3450-001  -1,0000+000

   4  8,3450-001   8,8074-001  -8,3450-001   1,0000+000   8,3450-001

   5  1,0000+000   9,9438-001  -1,0000+000   8,3450-001   1,0000+000
```

```
OVERALL MEANS AND VARIANCES BASED ON        14 TOTAL OBSERVATIONS

 VARIABLE        1            2            3            4            5
 MEAN       1,2143+001   1,9429+001   2,1429+001   2,7000+001   2,8214+001
 VARIANCE   6,7363+001   2,7503+002   3,3319+002   6,0185+002   8,4926+002



            1            2            3            4            5

   1  1,0000+000   9,4330-001   9,2378-001   9,0543-001   8,7013-001

   2  9,4330-001   1,0000+000   9,9012-001   9,9167-001   9,6863-001

   3  9,2378-001   9,9012-001   1,0000+000   9,9426-001   9,7563-001

   4  9,0543-001   9,9167-001   9,9426-001   1,0000+000   9,8353-001

   5  8,7013-001   9,6863-001   9,7563-001   9,8353-001   1,0000+000
```

continued

```
DISCRIMINANT FUNCTION NUMBER  1
DISCRIMINANT CRITERION FOR THIS FUNCTION  1;7242+002
PERCENT OF TRACE  9,2960+001
RAO-S CHI-SQUARE FOR THIS FUNCTION IS  5,1557+001 WITH     6 DEGREES OF FREEDOM
NORMED FUNCTION WEIGHTS

 VARIABLE       1           2           3           4           5
 WEIGHT     5,9922-001 -5,2618-001 -6,0280-001 -1,1874-002 -2,3633-002
STANDARDIZED FUNCTION WEIGHTS

 VARIABLE       1           2           3           4           5
 STD, WT,   3,3011-001 -5,8571-001 -7,3855-001 -1,9552-002 -4,6227-002
GROUP MEAN SCORES ON THIS FUNCTION

 GROUP          1           2           3
 MEAN SCR  -4,9457+000 -9,2613+000 -4,1220+001
```

```
DISCRIMINANT FUNCTION NUMBER  2
DISCRIMINANT CRITERION FOR THIS FUNCTION  1;3057+001
PERCENT OF TRACE  7,0398+000
RAO-S CHI-SQUARE FOR THIS FUNCTION IS  2,6431+001 WITH     4 DEGREES OF FREEDOM
NORMED FUNCTION WEIGHTS

 VARIABLE       1           2           3           4           5
 WEIGHT     9,0155-001 -4,3111-001  2,5524-002  2,0384-002 -1,6650-002
STANDARDIZED FUNCTION WEIGHTS

 VARIABLE       1           2           3           4           5
 STD, WT,   7,1677-001 -6,9257-001  4,5131-002  4,8440-002 -4,7001-002
GROUP MEAN SCORES ON THIS FUNCTION

 GROUP          1           2           3
 MEAN SCR   6,6098-001  6,0798+000  2,7706+000
```

continued

```
GROUP CENTROIDS IN DISCRIMINANT SPACE



           1           2           3

  1 -4,9457+000 -9,2613+000 -4,1220+001

  2  6,6098-001  6,0798+000  2,7706+000
```

```
INTERCENTROID DISTANCES IN DISCRIMINANT SPACE



           1           2           3

  1  0,0000+000  6,9274+000  3,6336+001

  2  6,9274+000  0,0000+000  3,2130+001

  3  3,6336+001  3,2130+001  0,0000+000
```

```
  0 OBSERVATIONS DROPPED FROM SCORES COMPUTATION  FOR  GROUP  1 DUE TO PARITY ERRORS
  0 OBSERVATIONS DROPPED FROM SCORES COMPUTATION  FOR  GROUP  2 DUE TO PARITY ERRORS
  0 OBSERVATIONS DROPPED FROM SCORES COMPUTATION  FOR  GROUP  3 DUE TO PARITY ERRORS
```

```
BACK SOLUTION OF DISCRIMINANT EQUATION



           1           2

  1 -1,7462-007  1,3970-008

  2 -5,4250-008  3,1665-008

  3  2,2631-007  9,8720-008

  4 -1,8254-007  1,3411-007

  5  9,2387-007  1.2293-007
```

COST ESTIMATE

For the job listed on the Sample Output, the total processing time was 17.007 seconds. At the current rate for the Michigan State University's CDC 3600 ($245/hr.), the computer time cost $1.16 plus a charge for input and output. A 10% surcharge will be assessed if billed to a non-University account. In addition, there is a consulting fee of $10./hr. for work done by the Applications Programming Group.

Charge to user = computer costs + consulting + network overhead
= $1.20 (approx.) + consulting + network overhead

CONTENTS—DISCRIM2

pages
1– 2 Identification & Abstract
3 User Instructions
5–10 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Nonmetric Multidimensional Scaling
CALLING NAME	KRUSKAL
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service (ETS)
AUTHOR(S) AND AFFILIATION(S)	Program due to: J.B. Kruskal R.F. Long Bell Telephone Laboratories Modification due to: D. Kirk Educational Testing Service
LANGUAGE	FORTRAN
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Office of Data Analysis Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

The latest version of the multidimensional scaling program written by J.B. Kruskal of Bell Telephone Labs is available. In addition to the improvements made from earlier versions, a much faster sort has been incorporated into the program and a multi-calculation facility using random starting configurations within one machine run, has been made available. This feature, in addition to being more efficient than the old technique of punching cards for subsequent runs, increases the chances of converging to a global minimum within one machine run with the corresponding savings in time and cost.

A modification to the output was made so that normally only the best of many possible configurations is printed. However, each final configuration and even the individual iterations may be printed (as was done in Kruskal's version) if desired.

The program will handle 60 subjects scaled in up to ten dimensions. Calculations can be repeated up to 99 times on as many as 1800 data values.

continued

000 0068

REFERENCES

Kruskal, J.B., "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," Psychometrika, *29*, pp. 1–27, (1964).

Kruskal, J.B., "Nonmetric Multidimensional Scaling: A Numerical Method," Psychometrika, *29*, pp. 115–129, (1964).

USER INSTRUCTIONS

Input

Program input may be broken down into five types of cards.

1. Control Cards governing the types of calculations desired
2. a Title Card identifying the run
3. a Parameter Card indicating size of matrix, replications, perturbations and print options
4. a Format Card identifying the form of the input data
5. the data

Control Cards

These are used, basically, to override preset program constants, if desired. They may be inserted in any order with the restriction that the "similarities" or "dissimilarities" option must be the last specified. They may be punched one per card or connected by commas. For a complete description see Ref. 1. The following may be used. (Punch # for = on an 029 keypunch.)

Control Options (begin in Col. 1)	*Explanation* (pre-set value is in parentheses)
DIMMAX = *integer*	Maximum dimension (2)
DIMMIN = *integer*	Minimum dimension (2)
DIMDIF = *integer*	Step size from maximum to minimum (1)
ITERATIONS = *integer*	Limitations for convergence (50)
DIAGONAL PRESENT DIAGONAL ABSENT	Data specification (present)
MATRIX HALFMATRIX	Data specification (matrix)
CUTOFF = *floating pt.*	Values $\leq$ cutoff are discarded (0.0)
WEIGHTS	Indicates weights for data follow (no weights)
R = *floating pt. > 1*	Metric space (2)
PRIMARY SECONDARY	Equal distance stress contribution (PRIMARY)

continued

Control Options (begin in Col. 1)	*Explanations* (pre-set value is in parentheses)
SFORM 1 SFORM 2	Different stress formulae (see Ref. 1) (SFORM 1) *Note:* Ref. 1 indicates that the standard option has proven superior in all cases tested.
NOCARDS CARDS	Final configuration punch option (NOCARDS) *Note:* When CARDS is requested, only the best configuration is punched.
CONFIGURATION	Identification of configuration deck (no configuration)
STRMIN = *floating point*	Satisfactory stress value (.01)
SFGRMN = *floating point*	Satisfactory gradient minimum (0.0)
SRATST = *floating point*	Satisfactory change in successive stress values (.999)
SIMILARITIES DISSIMILARITIES	Definition of data measurement (dissimilarities) *Note:* This option must be the last one specified.

Title Card

The Title Card may have any identification punched in Col. 1–72. It follows the SIMILARITIES or DISSIMILARITIES Card.

Parameter Card

The Parameter Card follows the Title Card.

Columns	*Contents*
1– 3	Size of data matrix
4– 6	No. of replications (i.e., No. of values for each matrix entry)
7– 9	No. of perturbations (i.e., No. of separate calculations with initial random configuration, ≤ 99)
10–12	1: print iteration values 0: do not print iteration values

continued

Columns	*Contents*
13–15	1: print final configuration for each perturbation 0: do not print final configuration

Format Card for Input Data

This is a regular Fortran format enclosed by parentheses,

Ex: (10 F 8.3)

A slight modification of this format is used to print the data as part of the output; care must be taken to assure that the entire number is printed. Since decimals on the Data Cards will override the format specifications, only the field width need be of concern when punching the Data Cards. However, data punched .821, .342, etc. with a format of (8 F 5.0) will read correctly but will print as zeros. With a format of F w.d, w should be at least three greater than the number of significant digits desired.

Data Cards

Data matrix input is *by rows* according to format specified on the Format Card. Begin each row on a new card, the count will be determined by the size of the data matrix. If a half matrix is being read, it is convenient to consider it as the lower triangular matrix, including or excluding the diagonal.

Following the last Data Card, a Control Card, COMPUTE, is placed. A Stop Control Card follows the last Compute Card for the entire data deck. Multiple cases may be handled on one machine run.

Compute Card

Columns	*Contents*
1– 7	COMPUTE

Stop Card

Columns	*Contents*
1– 4	STOP

System Control Cards

JOB Card will be prepared by ETS personnel.

continued

System Card 1

Columns	*Contents*
1–37	//STEPNAME EXEC GITNGO,NAME=KRUSKAL

System Card 2

Columns	*Contents*
1–19	//GO.SYSIN DD *

System Card 3

Columns	*Contents*
1– 2	//

Job Deck

Job Card
System Card 1
System Card 2
Data Deck
System Card 3

Output

The output is virtually self-explanatory. An added feature is the printout of the various control cards and the input data matrix so that visual checking for errors may be made.

Normally, the stress and best configuration are printed with a history of the stresses, number of iterations, and a convergence code for each perturbation. From rather limited experience it has been found that a minimum is generally located within ten perturbations (i.e., no smaller values were found when 40 or 50 were used).

When the CARDS punch option is requested, only the best configuration is punched.

Limitations

The present program will handle 60 objects scaled in up to ten dimensions. 1800 data values may be read, and up to 99 perturbations may be performed.

REFERENCES

1. Kruskal, J.B., "Nonmetric Multidimensional Scaling: A Numerical Method," Psychometrika, *29*, pp. 115–129, (1964).

SAMPLE INPUT

```
//KRT2   JOB (....
//STEPNAME  EXEC  GITNGO,NAME=KRUSKAL
//GO.SYSIN  DD    *
DIMMIN#1
ITERATIONS#50
SIMILARITIES
  TEST CASE WITH NEW WRITEUP
  5  1 10
(5F5.2)
9.3  0.57 6.1  2.0  1.5
1.0  8.5  5.5  3.3  4.0
2.1  5.3  7.3  3.9  1.1
5.0  6.2  1.6  8.9  2.8
1.3  4.1  1.0  2.5  8.8
COMPUTE
STOP
/*
//
```

SAMPLE OUTPUT

```
CONTROL CARDS
DIMMIN#1
ITERATIONS#50
SIMILARITIES
  TEST CASE WITH NEW WRITEUP
NO. OBJECTS#    5 REPLICATIONS#     1 PERTURBATIONS#
               10 ITERATION PRINT#  0 PERTURBATION PRINT#  0

9.30  0.57  6.10  2.00  1.50
1.00  8.50  5.50  3.30  4.00
2.10  5.30  7.30  3.90  1.10
5.00  6.20  1.60  8.90  2.80
1.30  4.10  1.00  2.50  8.80
COMPUTE


THE BEST CONFIGURATION OF   5 POINTS IN  2 DIMENSIONS
                                       HAS STRESS  0.037

BEST CONFIGURATION
     1      2
 1 -1.052 -0.677
 2  0.529  0.769
 3 -0.750  0.588
 4  0.059 -0.350
 5  1.215 -0.300

              HISTORY OF PERTURBATIONS
         CODE      STRESS      ITERATIONS
          2.0       0.225            20.0
          2.0       0.230            20.0
          2.0       0.184            23.0
          2.0       0.037            24.0
          2.0       0.037            26.0
          2.0       0.037            26.0
          2.0       0.037            32.0
          2.0       0.037            30.0
          2.0       0.037            34.0
          2.0       0.037            30.0

 CODES--%1<ZERO STRESS,%2<MINIMUM ACHIEVED,
     %3<SATISFACTORY STRESS REACHED,%4<MAXIMUM ITERATIONS USED

THE BEST CONFIGURATION OF   5 POINTS IN  1 DIMENSIONS
                                       HAS STRESS  0.317

BEST CONFIGURATION
     1
 1 -1.493
 2  1.451
 3 -0.042
 4 -0.533
 5  0.617

              HISTORY OF PERTURBATIONS
         CODE      STRESS      ITERATIONS
          2.0       0.409            17.0
          2.0       0.413            25.0
          2.0       0.381            22.0
          2.0       0.318            20.0
          2.0       0.317            27.0
          2.0       0.447            15.0
          2.0       0.399            15.0
          2.0       0.378            17.0
          2.0       0.423            16.0
          2.0       0.367            20.0

 CODES--%1<ZERO STRESS,%2<MINIMUM ACHIEVED,
     %3<SATISFACTORY STRESS REACHED,%4<MAXIMUM ITERATIONS USED

 **********************************************
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $3.66.

Charge to user = computer costs + postage and handling + network overhead
= $3.66 + $5.00 + network overhead
= $8.66 + network overhead

CONTENTS—KRUSKAL

pages

1– 2	Identification & Abstract
3– 6	User Instructions
7– 8	I/O
9	Cost—Contents

DESCRIPTIVE TITLE	Plot
CALLING NAME	PLOT
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service (ETS)
AUTHOR(S) AND AFFILIATION(S)	University of Michigan
LANGUAGE	MAP
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Off. of Data Anal. Research, Educational Testing Service Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

A flexible plot routine originally written for the 7090 at the University of Michigan and adapted to their system, has been modified for the operating system at ETS. Although the program is written in assembly language, the necessary linkage to FORTRAN and FORTRAN I/O has been written. It is intended to be called, therefore, by a FORTRAN program and will produce plots in the normal output stream as determined by the various calls, to different entry points in the subroutine. The standard approach is to call PLOT1, PLOT2, PLOT3, and PLOT4 in that order. PLOT1 sets up the information required to construct the graph, PLOT2 prepares the grid and sets up the information required by PLOT3 to place points in the graph, PLOT3 places the plotting characters at the specified points in the graph, and PLOT4 prints the completed graph with values along the X and Y axes and a centered vertical label down the left side of the graph.

USER INSTRUCTIONS

PLOT1

PLOT1 sets up the information required to construct the graph. It consists of five arguments, the first of which is the first location of a five-word vector. The calling sequence is

```
CALL PLOT1 (NSCALE(1), NHL, NSBH, NVL, NSBV)
```

The five arguments are defined as follows.

NSCALE(1)—controls the scaling and number of decimals printed. If this is 0, the values 0, 3, 0, 3 are used for NSCALE(2) through NSCALE(s), respectively.

NSCALE(2)—if this has the value n, the numbers printed along the y axis are 10^n times their true value.

NSCALE(3)—the number of decimal places printed for y values.

NSCALE(4)—if this has the value n, the numbers printed along the x axis are 10^n times their true value.

NSCALE(5)—the number of decimal places printed for X values.

NHL—number of horizontal lines—must be two or greater.

NSBH—number of spaces between horizontal lines—must be one or greater.

NVL—number of vertical lines—must be two or greater.

NSBV—number of spaces between vertical lines—must be one or greater.

NOTE: If different values for NSCALE(1) through NSCALE(5) are desired, then the call to NSCALE(1) should be the name of a five-entry array containing the desired values.

PLOT2

PLOT2 prepares the grid and sets up the information required by PLOT3 to place a point correctly in the graph. The calling sequence is

```
CALL PLOT2 (IMAGE, XMAX, XMIN, YMAX, YMIN)
```

If PLOT1 has not been called prior to PLOT2, the following default configuration is used: NSCALE(1) = 0, NHL = 6, NSBH = 9, NVL = 11, NSBV = 9. The five arguments are defined as follows.

IMAGE—the region used to hold the image of the graph. A dimension of 2000 should be sufficient.

continued

XMAX—largest X value to be plotted.

XMIN—smallest X value to be plotted.

YMAX—largest Y value to be plotted.

YMIN—smallest Y value to be plotted.

PLOT3

PLOT3 places the plotting character in the graph for each point (X,Y). Its calling sequence is

```
CALL PLOT3 (BCD, X, Y, NDATA, INT)
```

The five arguments are defined as follows.

BCD—plotting character used. Example: (1H*).

X—the first element of a floating point vector of X values.

Y—the first element of a floating point vector of Y values.

NDATA—the number of points to be plotted.

INT—the number of bytes between the characteristics of numbers used as coordinates (short = 4, long = 8).

PLOT4

PLOT4 will print the completed graph with values along the X and Y axes with a centered vertical label down the left side. The calling sequence is

```
CALL PLOT4 (NCHAR, LABEL)
```

The two arguments are defined as follws.

NCHAR—the number of characters in the label.

LABEL—the location of the first character of the label. The following example is representative.

```
CALL PLOT4 (18, 'ANY VERTICAL LABEL')
```

System Control Cards

JOB Card will be supplied by ETS.

System Card 1

Columns	*Contents*
1–22	//PROG EXEC FORTGCLC,

continued

System Card 2

Columns	*Contents*
1–32	// REGION.GO=120K,ACCT.GO=0200

System Card 3

Columns	*Contents*
1–18	//FORT.SYSIN DD *

System Card 4

Columns	*Contents*
1–36	//LKED.LIB DD DSN=OCS.SAPGM,DISP=SHR

System Card 5

Columns	*Contents*
1–19	//LKED.SYSIN DD *

System Card 6

Columns	*Contents*
11–27	INCLUDE LIB(PLOT)

System Card 7

Columns	*Contents*
1– 2	/*

System Card 8

Columns	*Contents*
1– 2	//

Job Deck

JOB Card
System Card 1
System Card 2
System Card 3
Input Deck
System Card 4
System Card 5
System Card 6
System Card 7
System Card 8

continued

1-8

The preceding information should suffice for a high percentage of the graphs required. A subroutine has been written to insert BCD information in the body of the graph, and additional features exists to handle semi-log and log-log plots, but have not been completely checked out as yet. Anyone wishing to use these options should consult the contact person.

SAMPLE INPUT

EXAMPLE 1

```
//  (JOB CARD)
//PROG   EXEC FORTGCLG,
//   REGION.GO=120K,ACCT.GO=02000
//FORT.SYSIN  DD *
       DIMENSION X1(100),X2(100),X3(100).GRAPH(2000),IN(5)
       REAL*8  GRAPH2(1000)
       COMMON GRAPH,NVL,NSBV
       EQUIVALENCE(GRAHP,GRAPH2)
       DIMENSION A(100)
       NDATA = 100
       DO 2 I=1,100
       A(I) = I
       X1(I) = A(I)/100.
       X2(I) = 1.-X1(I)
2       X3(I) = X1(I) * X1(I)
       NHL=6
       NSBH=9
       NVL=6
       NSBV=19
       IN(1)=1
       IN(2)=0
       IN(3)=1
       IN(4)=0
       IN(5)=1
       CALL PLOT1 (IN,NHL,NSBH,NVL,NSBV)
       CALL PLOT2(GRAPH,100.,0.,1.,0.)
       CALL PLOT3(1H*,A,X1,NDATA,4)
       CALL PLOT3(1H",A,X2,NDATA,4)
       CALL PLOT3(1H#,A,X3,NDATA,4)
       CALL INSH(18,6,5,'MACHINE AVAILABILITY')
       CALL INSH(9,35,5,'PROGRAMMING DEMANDS ')
       CALL INSH(40,45,4,'BLOOD PRESSURE   ')
       WRITE (6,5)
5      FORMAT (1H1,40X,'SO YOU THINK YOU HAVE TROUBLES-----')
       CALL PLOT4(40,'READ THIS V/E/R/Y  CAREFULLY                ')
       WRITE (6,8)
 8      FORMAT(1H0,40X,'WILL THESE SOON BE REVERSED?',50X,'DBK')
       END
//LKED.LIB DD DSN=OCS.SAPGM,DISP=SHR
//LKED.SYSIN DD *
               INCLUDE LIB(PLOT)
       INCLUDE LIB(INSH)
/*
//
```

continued

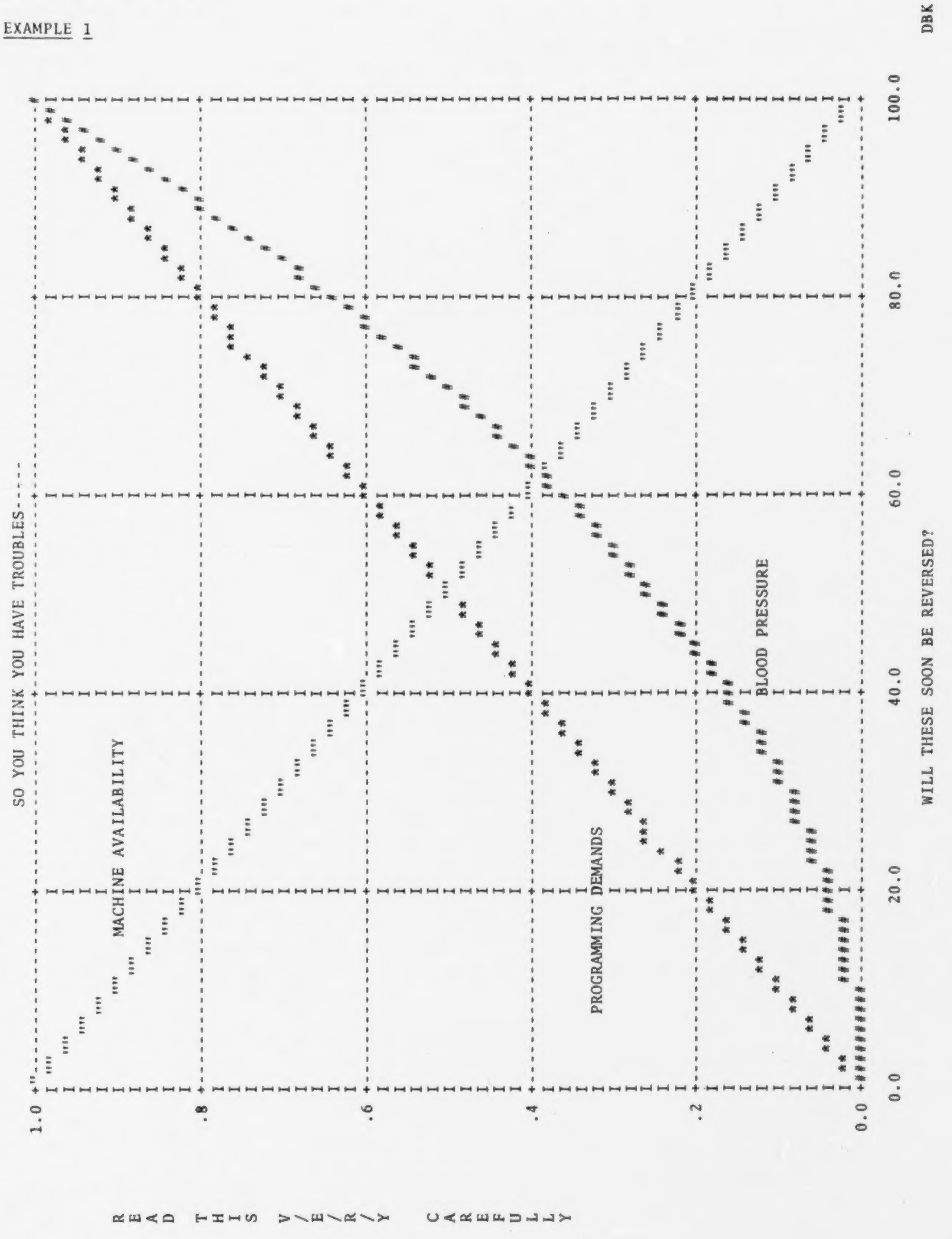
EXAMPLE 2

Control cards are not listed in this example.

```
         REAL*8 GRAPH2(1000)
         EQUIVALENCE (GRAPH,GRAPH2)
         DATA BLANK/'     '/,BCD/'*'/
         POL(ANGLE) =.25 * ANGLE
         RMAX = 4.
         NSBV=79
         NVL=2
         CALL PLOT1 (0,2,47,2,79)
         CALL PLOT2 (GRAPH,RMAX,-RMAX,RMAX,-RMAX)
         DO 1 I=1,2000
1         GRAPH(I)=BLANK
         DELTAY = RMAX/24.
         DELTAX=RMAX/40.
         IMAX = 2.*RMAX+1.
         DO 4 I=1,49
4         CALL PLOT3(1HI,0.,RMAX-FLOAT(I-1)*DELTAY,1,4)
         DO 5 I=1,IMAX
5         CALL PLOT3(1H+,0.,RMAX-FLOAT(I-1),1,4)
         DO6 I=1,81
6         CALL PLOT3(1H-,-RMAX+FLOAT(I-1)*DELTAX,0.,1,4)
         DO 7 I=1,IMAX
7         CALL PLOT3(1H+,-RMAX+FLOAT(I-1),0.,1,4)
         DO 12 I=1,140
         THET = .1 * FLOAT(I)
         SWITCH = PLOT3(BCD,POL(THET)*COS(THET),POL(THET)*SIN(THET),1,4)
12          CONTINUE
 13      CALL OMIT(3)
         CALL INSH(44,1,3,'90 DEGREES  ')
         CALL INSH(2,26,3,'180 DEGREES ')
         CALL INSH(71,26,3,' 0 DEGREES  ')
         CALL INSH (44,47,3,'270 DEGREES ')
          WRITE (6,9)
9         FORMAT(1H1,40X,' A SLIGHTLY DIFFERENT EXAMPLE')
         CALL PLOT4(26,'POLAR PLOT OF THE SPIRAL               ')
         END
```

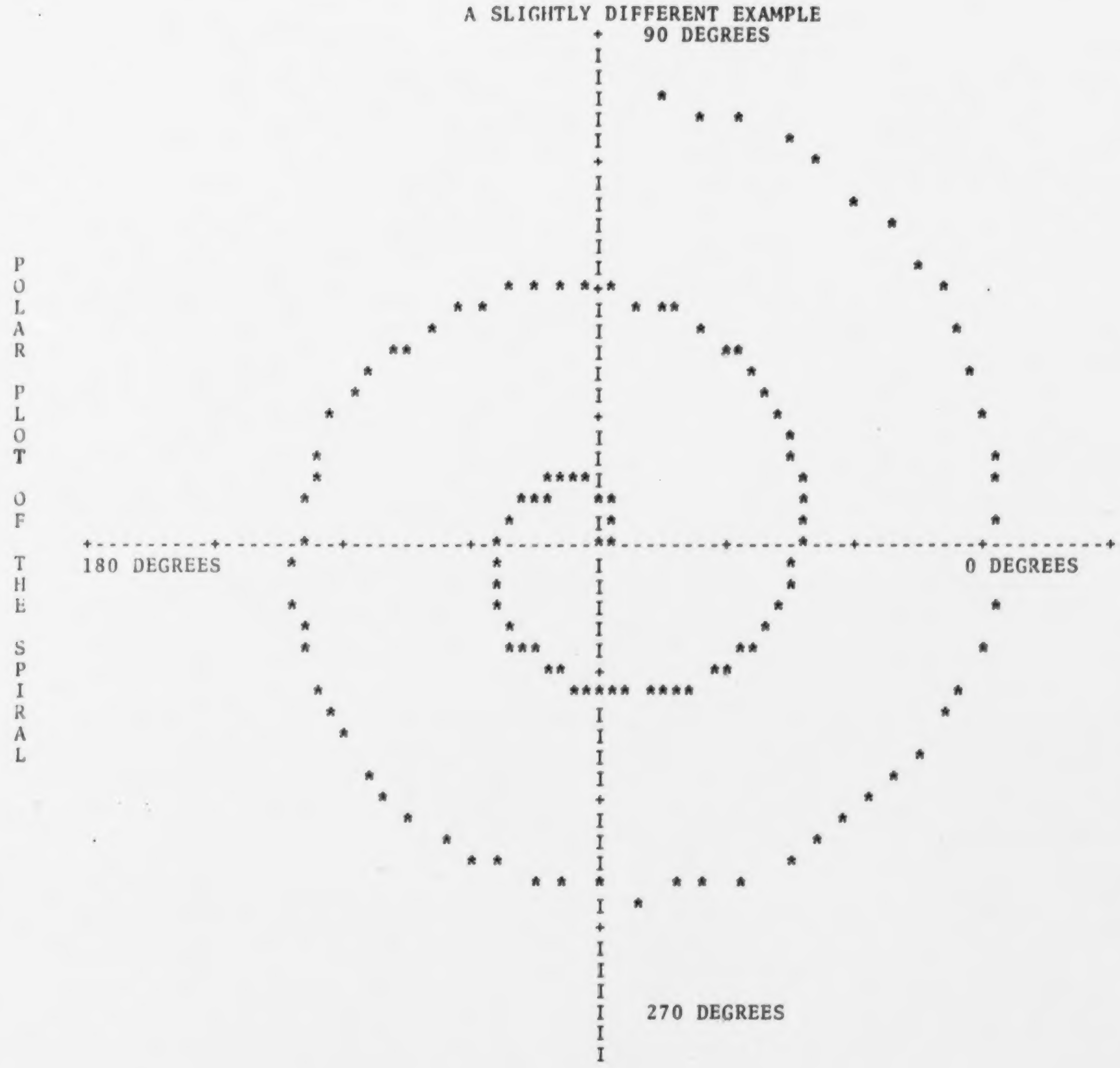
SAMPLE OUTPUT

EXAMPLE 1



continued

EXAMPLE 2


A SLIGHTLY DIFFERENT EXAMPLE
90 DEGREES
180 DEGREES
0 DEGREES
270 DEGREES
POLAR PLOT OF THE SPIRAL

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for Example 1 in the Sample Input were $2.01.

Example 1

Charge to user = computer costs + postage and handling + network overhead
= $2.01 + $5.00 + network overhead
= $7.01 + network overhead

Computer costs for Example 2 in the Sample Input were $2.09.

Example 2

= $2.09 + $5.00 + network overhead
= $7.09 + network overhead

CONTENTS—PLOT

pages

1 Identification & Abstract
3– 6 User Instructions
7–10 I/O
11 Cost—Contents

DESCRIPTIVE TITLE	Multidimensional Scalogram Analysis
CALLING NAME	MSA-I
INSTALLATION NAME	Office of Data Analysis Research Educational Testing Service
AUTHOR(S) AND AFFILIATION(S)	Procedure due to: L. Guttman J. Lingoes University of Michigan Program due to: J. Lingoes University of Michigan Adaptation due to: D. Kirk Educational Testing Service Advisor on use: H. Harman Educational Testing Service
LANGUAGE	MAP
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Off. of Data Anal. Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

MSA-I is a program to map types (individuals having the same profile over a set of variables or items) onto an Euclidean space with minimum dimensionality. No assumptions are required about the underlying distributions, the scaling properties of the items, or their ordering. The only requirement is that the categories of each item be mutually exclusive and exhaustive. Types are represented as points in space, each item is a partition of the space, and each category is a region. All types who fall in a given category of a particular item are constrained to be closer to boundary markers of the same category than to delimiters of other categories of the same item. The program we have is a modification of the MSA-I package as distributed

continued

by Dr. Lingoes. An option or modification provides additional plots for each category with the number inserted in the plots corresponding to the response to the item rather than the ID number as in the distributed version.

References and some copies of the substantial literature related to this problem are available through the contact person.

REFERENCE

Lingoes, J.C., "Multiple Scalogram Analysis—A Set-theoretic Model for Analyzing Dichotomous Items," J. of Ed. & Psych. Meas., *23*, pp. 501-524, (1963).

USER INSTRUCTIONS

Input Deck

Title Card

Columns	*Contents*
1–80	Title information

Control Card (zeros and blanks treated alike)

Columns	*Contents*
1– 4	Number of variables ($\leq$ 50)
5– 8	Number of subjects ($\leq$ 100)
9–12	Precoded switch 1: values are integers in the range 1–20 0 or blank: values are not in this range
13–16	If precoded switch is 1, leave this blank If precoded switch is 0 or blank, punch number of intervals into which data are to be coded.
17–20	Smallest frequency to be permitted after coding. Blank indicates no lower limits are desired.
21–24	1: card output of coded data is desired 0: card output is not desired
25–28	Number of types of nonredundant profiles. If program is to determine redundancies, set this item to the number of subjects.
29–32	Minimum number of dimensions (default is 1)
33–36	Maximum number of dimensions (either 2 or 3)
37–40	Number of iterations (default is 25)
41–44	1: item plots for each item 0: no item plots It is suggested that this be set to 1.
45–48	Cutoff criterion for coefficient of contiguity. Floating point ($0 < c \leq 1.00$)
49–52	Code to indicate mean is to be substituted for missing data.

continued

Frequency Card Deck (Use if Number of types of nonredundant profiles ≠ 0.

Columns	*Contents*
1– 2	Frequency for first type
3– 4	Frequency for second type
⋮	⋮
71–72	Frequency for thirty-sixth type

If there are more than 36 types, continue onto another card.

Format Card

Data Deck

Multiple cases may be run, each with its own Input Deck.

Job Deck

The program is currently available and may be executed as follows.

```
//JOB card (will be provided by ETS personnel)
//EXEC GITNGO,NAME=MSAX
//GO.SYSIN DD *
(Insert Input Deck or Decks here)
/*
```

Output

The output will depend upon the options selected in the Control Card.

SAMPLE INPUT

```
//DKMSA   JOB (....
//EXEC GITNGO,NAME=MSAX
//GO.SYSIN DD *
1TEST FOR G-L(MSA-I) FOR A PERFECT UNIDIMENSIONAL SCALE OF 3
                                                  CATAGORIES.
   3   7       3           7   2   2       1 1.0
 4 1 1 2 1 1 1
(3F1.0)
333
332
322
222
221
211
111
/*
```

000 0070

SAMPLE OUTPUT

```
TEST FOR G-L(MSA-I) FOR A PERFECT UNIDIMENSIONAL SCALE OF 3
                                                  CATAGORIES.
 VARIABLE          SCORE   RANGE          CODED INTERVAL
----------------------------------- ----------------------

     1     0.100000E 01 TO 0.300000E 01    0.100000E 01
     2     0.100000E 01 TO 0.300000E 01    0.100000E 01
     3     0.100000E 01 TO 0.300000E 01    0.100000E 01

        FREQUENCY DISTRIBUTION FOR RANKED CODED DATA

VAR.      1     2     3     4     5     6     7     8     9
------------------------------------------------------------

   1      1     3     3
   2      2     3     2
   3      3     3     1

SCORES FOR EACH SUBJECT ON EACH VECTOR

VECTOR          1     2     3
-------------------------------
SUBJECT
     1       -529.  509.
     2       -409.   69.
     3       -230. -320.
     4          0. -517.
     5        230. -320.
     6        409.   69.
     7        529.  509.
```

```
      LINGOES MULTIVARIATE ANALYSIS OF CONTINGENCIES

CATEGORY WEIGHTS
----------------

VECTOR   1.     ETA =.811
CATEGORY         1     2     3     4     5     6     7     8
-------------------------------------------------------------

VARIABLE
    1        529.  213. -389.
    2        469.    0. -469.
    3        389. -213. -529.

VECTOR   2      ETA =.368
CATEGORY         1     2     3     4     5     6     7     8
-------------------------------------------------------------
VARIABLE
    1        509. -256.   86.
    2        289. -385.  289.
    3         86. -256.  509.
```

continued

```
GUTTMAN-LINGOES OUTER-POINT SCALOGRAM ANALYSIS COORDINATES
                                                FOR M = 2.
DIMENSION          1          2
------------------------------------------------------------


     TYPE
       1   -100.000     30.643
       2    -77.180    -25.472
       3    -43.381    -74.921
       4      0.000   -100.000
       5     43.382    -74.921
       6     77.180    -25.471
       7    100.000     30.644

COEFFICIENT OF CONTIGUITY =0.100000E 01 FOR     1 ITERATIONS.




OUTER-POINT MATRIX
-------------------


 TYPE =  1 ( 4)
       3   3  -3

 TYPE =  2 ( 1)
       3  -3  -2

 TYPE =  3 ( 1)
      -3  -2   2

 TYPE =  4 ( 2)
      -2   2  -2

 TYPE =  5 ( 1)
       2  -2  -1

 TYPE =  6 ( 1)
      -2  -1   1

 TYPE =  7 ( 1)
      -1   1   1
```

continued

```
VECTOR PLOTS
VECTOR    2 PLOTTED AGAINST VECTOR    1
                                                            VECTOR
                                                               2

         -100  -90  -80  -70  -60  -50  -40  -30  -20  -10  *    10   20   30   40   50   60   70   80   90  100
          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      100 *                                                 *                                                 * 100
       96 *                                                 *                                                 * 96
       92 *                                                 *                                                 * 92
       88 *                                                 *                                                 * 88
       84 *                                                 *                                                 * 84
       80 *                                                 *                                                 * 80
       76 *                                                 *                                                 * 76
       72 *                                                 *                                                 * 72
       68 *                                                 *                                                 * 68
       64 *                                                 *                                                 * 64
       60 *                                                 *                                                 * 60
       56 *                                                 *                                                 * 56
       52 *                                                 *                                                 * 52
       48 *                                                 *                                                 * 48
       44 *                                                 *                                                 * 44
       40 *                                                 *                                                 * 40
       36 *                                                 *                                                 * 36
       32 * 1                                               *                                               7 * 32
       28 *                                                 *                                                 * 28
       24 *                                                 *                                                 * 24
       20 *                                                 *                                                 * 20
       16 *                                                 *                                                 * 16
       12 *                                                 *                                                 * 12
        8 *                                                 *                                                 * 8
        4 *                                                 *                                                 * 4
VECTOR  1 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       -4 *                                                 *                                                 * -4
       -8 *                                                 *                                                 * -8
      -12 *                                                 *                                                 * -12
      -16 *                                                 *                                                 * -16
      -20 *                                                 *                                                 * -20
      -24 *                                                 *                                                 * -24
      -28 *        2                                        *                                     6           * -28
      -32 *                                                 *                                                 * -32
      -36 *                                                 *                                                 * -36
      -40 *                                                 *                                                 * -40
      -44 *                                                 *                                                 * -44
      -48 *                                                 *                                                 * -48
      -52 *                                                 *                                                 * -52
      -56 *                                                 *                                                 * -56
      -60 *                                                 *                                                 * -60
      -64 *                                                 *                                                 * -64
      -68 *                                                 *                                                 * -68
      -72 *                                                 *                                                 * -72
      -76 *                          3                      *                   5                             * -76
      -80 *                                                 *                                                 * -80
      -84 *                                                 *                                                 * -84
      -88 *                                                 *                                                 * -88
      -92 *                                                 *                                                 * -92
      -96 *                                                 *                                                 * -96
     -100 *                                                4*                                                 *-100
          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         -100  -90  -80  -70  -60  -50  -40  -30  -20  -10  *    10   20   30   40   50   60   70   80   90  100
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $2.67.

Charge to user = computer costs + postage and handling + network overhead
= $2.67 + $5.00 + network overhead
= $7.67 + network overhead

CONTENTS—MSA-I

pages
1– 2 Udentification & Abstract
3– 4 User Instructions
5– 8 I/O
9 Cost—Contents

DESCRIPTIVE TITLE	Simulation Package for University Research and Training
CALLING NAME	SPURT
INSTALLATION NAME	Vogelback Computing Center Northwestern University
AUTHOR(S) AND AFFILIATION(S)	Gustave J. Rath Department of Industrial Engineering and Management Sciences Martin Goldberg Leonard Weiner Northwestern University
LANGUAGE	CDC FORTRAN IV
COMPUTER	CDC 6400
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Lorraine Borman, EIN Technical Representative, Vogelback Computing Center, Northwestern University, 2129 Sheridan Road, Evanston, Ill. 60201 Tel.: (312) 492-3682

FUNCTIONAL ABSTRACT

SPURT is a comprehensive package of USASI Standard FORTRAN routines that are designed for use in simulation modelling. These useful routines, ranging from simple to complex, enable the average FORTRAN programmer to employ simulation techniques without having to learn the semantic and syntactic rules of a new programming language.

The SPURT package is made up of six main parts.

I. *CLOCK Generation—SPURT1*
II. *Stochastic Generators—SPURT2*
III. *Statistical Computations—SPURT3*
IV. *Analog Simulators—SPURT4*
V. *List-Processing and Queue-Manipulation—SPURT5*
VI. *Matrix and Graphical Output—SPURT6*

continued

The following is a listing and brief discussion of the various subroutines contained in each of the six SPURT parts.

CLOCK Generation—SPURT1: to implement discrete-time simulation models; to cause events to occur in the proper time sequence

The CLOCK subroutine consists basically of two lists:

Master Time List—contains events scheduled to happen in the future

Master Time Queue—contains events that could not take place at the time when they were scheduled to and, therefore, have been rescheduled; i.e., they have been blocked and are waiting in a queue.

Events can be stored on either list.

CLOCK recognizes two basic kinds of events:

Exogenous—those that are *external* to the user's routine; these are read from Data Cards by the CLOCK

Endogenous—those that are *internal* to the user's routine; these are generated dynamically and then are maintained by the CLOCK

Stochastic Generators—SPURT2: to generate samples from various probability distributions and to calculate sample values

Subroutine	*Usage*
STOGN1	Permits sampling from a discrete empirical probability distribution defined by the user
STOGN2	Enables the user to approximate a continuous distribution by means of a piecewise linear distribution
UNIFRM	Permits the user to sample real values from a uniform distribution in a defined interval
RANDIN	Provides a uniform distribution of integers in a defined interval
NORMAL	Allows the user to obtain a random sample from a normal distribution with given mean and standard deviation
NEGEXP	Permits the user to obtain a random sample from the negative exponential distribution
POISSN	Provides the user with a random sample from the Poisson distribution

continued

ERLANG	Provides a random sample from the Erlang distribution
DISCRT	Permits sampling from a step function describing a discrete cumulative distribution of integer values
LINEAR	Provides the user with a random sample from a cumulative distribution that is obtained by linear interpolation in a nonequidistant table of real values
DRAW	Provides a boolean value of TRUE or FALSE
RANPER	Generates a uniformly distributed, random permutation of the integers 1, 2, ···, M

Statistical Computations—SPURT3: to calculate statistical parameters and histograms of data arrays

Subroutine	*Usage*
STIX1 STIX2 STIX3	Three interrelated subroutines to accumulate and print out a frequency table and to produce a CalComp plot of a normalized histogram of the table
STIX4	Evaluates the mean, standard deviation, maximum value, and minimum value of an array of real numbers
STIX5	Evaluates the correlation coefficient between two arrays of real numbers
STIX6	Ranks an array of real numbers and produces the median and range of the data within the array
STIX7	Produces a statistical description of the data found in an array, including the sample size, mean, standard deviation, standard error, minimum and maximum values, range, and a printed histogram plot

Analog Simulators—SPURT4: to enable the simulation of analog-computer problems on a digital computer

Subroutine	*Usage*
ANALOG SECND	These two subroutines make it possible to obtain output similar to a hybrid computer

continued

List-Processing and Queue-Manipulation—SPURT5: lists are mXn arrays; entries in lists are mX1 arrays.

Subroutine	*Usage*
ADFIFO	Adds an entry at the bottom of the list; it can be removed only after all the elements presently on the list are gone (builds first-in last-out list)
ADLIFO	Adds an entry at the top of the list; it will be removed before any other entry presently on the list (builds last-in first-out list)
REMOVE	Removes the top (or first) entry from a list
PURGE	Destroys the contents of a list
DISPL	Prints the contents of a list

Additional subroutines in SPURT5 provide the capability to rank lists and to delete or to insert entries into lists.

Matrix and Graphical Output—SPURT6: output is facilitated through printing and graphical output

Subroutine	*Usage*
OUT	Prints out a square matrix with column and row headings
NSOUT	Prints out a nonsquare matrix with column and row headings
GRAPH	Produces two-dimensional graphs of plots, using a CalComp plotter

REFERENCES

1. Rath, G.J., "Description of the Simulation Package for University Research and Teaching," Northwestern Univ. Vogelback Computing Ctr., SPURT, Rev. E (1968; unpublished).

USER INSTRUCTIONS

SPURT is separated into six sections (Parts) in the Vogelback Program Library. All of the routines contained in any one Part will be loaded with the user's program when he places that Part name in the argument list of a Library Card. The following listing shows which routines are included in each part of SPURT.

Program Library Name (Part)					
SPURT1	SPURT2	SPURT3	SPURT4	SPURT5	SPURT6
SPURT User Routines Loaded:					
CLOCK	STOGN1	OUT	ANALOG	ADFIFO	TIMELP
SETCLK	STOGN2	IOUT	SECND	ADLIFO	PAGE
ANALOG	STOGN6	NSOUT		REMOVE	DIAGNS
SECND	UNIFRM	INSOUT		PURGE	DIAGNW
	RANDIN	GRAPH		DISPL	
	NORMAL	STIX1		IDISPL	
	NEGEXP	STIX2		NUMLST	
	POISSN	STIX3		RANKQS	
	ERLANG	STIX4		INRKQS	
	DISCRT	STIX5		RANKQL	
	LINEAR	STIX6		INRKQL	
	DRAW	STIX7		SEARCH	
	HISTD			INSERT	
	RANPER			DELETE	
SPURT Internal Routines Loaded:					
ZZXXQ9Q		QRLQPB			BLKDATU
LODLST					
PURGER					
Common Blocks Used:					
CLOCK1		SPRTFT		SZSYSP	QPLOTQ
SPRTIO		SPRTIO		SPRTFT	SZSYSP
		SZSYSP		SPRTIO	SPRTFT
		QPLOTQ			SPTMSG
		SPRTPL			SPRTIO
					QPAGEQ
SPURT Part Core Requirements (octal):					
6500	4000	21300 + 6400		5700	5000
Part Core Requirements When Loaded with SPURT6 (octal):					
7600	6600	22500 + 7600		7100	—

continued

NOTE: Whenever any of the SPURT routines is called from the library, *SPURT6 must also be called.* Therefore, *every* use of the Library Card for SPURT will include SPURT6 in the argument list.

Application Hints

All user-written subroutines should have as their first executable instruction a call to the user-written subroutine TRACE, passing as a parameter the name of the calling subroutine. For example, the first executable statement in subroutine FACTORY, say, should be CALL TRACE (7HFACTORY). Thus, as each subroutine within the simulation model initiates execution, it will call on the TRACE routine, which can be constructed to print a short message giving the current simulated time, the name of the calling subroutine, and the status of key system variables.

It should be emphasized that TRACE is a user-written closed subroutine. Therefore, it is the user's responsibility to construct the printout to include the simulation time, calling subroutine name, and all key system variables.

Segment coding is suggested because discrete-time simulation models are characterized by a chain of events and/or activities. Each event and activity is well defined in setting up the problem solution. This implies that each event or activity within the model should be coded as a closed subroutine. In doing this, it is good practice to place into a common block all variables used by more than one routine. Segmentation of this sort will also allow the easy generation of closed subroutines to handle data input, statistical analysis, printouts, and facilities for a TRACE routine.

Error-Indicator Subroutines

DIAGNS and/or DIAGNW

```
SUBROUTINE DIAGNS (NAME,NUMBR)
SUBROUTINE DIAGNW (NAME,NUMBR)
```

where NUMBR is an integer and NAME is a Hollerith constant.

The user may wish to incorporate calls to DIAGNW and/or DIAGNS in his main program. These are called by most of the SPURT subroutines and functions whenever an error condition arises during a run. If the error is of a serious nature, DIAGNS is called; it causes a program termination after an appropriate diagnostic message has been printed on the standard output unit. For less serious errors, DIAGNW is called by the SPURT routine detecting the error; control is returned to the calling routine after DIAGNW

continued

prints out a message. (See Ref. 1 for a listing of all diagnostic messages.)

The user is given the ability to force SPURT to continue execution of a program after detecting a normally fatal SPURT error condition during the run. To continue, the user *must preset the third word of* COMMON/SZSYSP *to nonzero in his main program*. This causes all calls (by SPURT) to DIAGNS to be handled as calls to DIAGNW.

CAUTION: The disablement of the termination feature of DIAGNS will merely cause an incorrect value to be stored in X; however, with many other SPURT routines (e.g., CLOCK), *infinite loops may occur when this feature is disabled. This option is to be used with extreme care!*

REFERENCES

1. Rath, G.J., "Description of the Simulation Package for University Research and Teaching," Northwestern Univ. Vogelback Computing Ctr., SPURT, Rev. E (1968; unpublished).

USER INSTRUCTIONS—SPURT1: CLOCK Generation

SUBROUTINE CLOCK (L,NEWT,NEWJ)
SUBROUTINE SETCLK (MTL,JUMP,LMTL,MTQ,JUMQ,LMTQ)
TYPE INTEGER L,NEWT,NEWJ,MTL,JUMP,LMTL,MTQ,JUMQ,LMTQ

The primary function of CLOCK is the manipulation of two lists: a Master Time List and a Master Time Queue. Events and their type number and scheduled time of occurrence can be stored on either list. The Master Time List contains events that are scheduled to happen; the Master Time Queue contains events that could not take place at the time when they were originally scheduled to and, therefore, have been rescheduled. That is, they were blocked and so are waiting in a queue. CLOCK causes events to occur at the proper simulation time. (Unless otherwise noted, all references to *time* here will mean *simulated time*.)

The Master Time List consists of two one-dimensional arrays: MTL and JUMP. Both are of the same size and must be dimensioned in the user's program. The MTL array contains the time at which the events are scheduled to occur. The JUMP array contains the number of the event type. The ith element of MTL and the ith elements of JUMP constitute an event. LMTL stands for the length of the Master Time List (LMTL $\leq$ that of MTL). NEVEN is the variable that contains the surrent number of events in the Master Time List.

NOTE: The value of NEVEN is maintained by the CLOCK and should not be changed by the user. It is intended for use on a *read-only* basis.

The Master Time Queue is very similar to the Master Time List, except that it contains blocked events. The variables MTQ, JUMQ, LMTQ, and NEVEQ correspond exactly to the variables MTL, JUMP, LMTL, and NEVEN for the Master Time List. Thus, MTQ is a one-dimensional array containing event types. Both arrays are dimensioned by the user in his program and must have the same dimension. Also, LMTQ must be defined (LMTQ $\leq$ dimension of MTQ). NEVEQ is the current number of events in the Master Time Queue.

CLOCK recognizes two basic kinds of events: exogenous (external to the user's program and the CLOCK subroutine) and endogenous (internal to the program).

Exogenous events are read in on data cards as input data, one event per card. The data cards must be arranged by the user in time sequence. To have SETCLK or CLOCK read an exogenous event, the user must specify the logical unit number (LUN) from which the cards are

continued

to be read. If the data cards follow the program on the standard system input, LUN = 5. If the value for LUN is not set between 1 and 99, the reading of exogenous events is suppressed.

Endogenous events are not read in on data cards and are created in the user's program.

The basic function of CLOCK is to cause events to happen when they are scheduled. When CLOCK is initialized, the first exogenous event is read (if a valid LUN has been specified) and is placed on the Master Time List. Subsequently, whenever the normal CALL CLOCK statement is used, the subroutine causes the event to be taken from the Master Time List or Master Time Queue, whichever has the earliest time of occurrence (ties are decided in favor of the Master Time Queue) and returned to the user's program. Whenever an exogenous event is deleted from the Master Time List, the next exogenous event is automatically read (if LUN is valid) and placed on the Master Time List in sequence. CLOCK works with only one event at a time and stops when all exogenous events have been read in and both lists are empty.

The user initializes CLOCK with the following call: CALL SETCLK(MTL, JUMP,LMTL,MTQ,JUMQ,LMTQ). The parameters in the calling sequence are the same as those described above.

If the user wishes to read an exogenous event, then common block CLOCK1 must be accessed (see Ref. 1, p. 3.9) and LUN preset equal to 5 (see Ref. 1, p. 3.4).

The user calls CLOCK with CALL CLOCK(L,NEWT,NEWJ). The only variables that will change in the course of the user's program are L, NEWT, and NEWJ. Thus, SETCLK is called only when it is desired to initialize both clock lists. Here, NEWT contains the time of the event and NEWJ contains the number of the event type. Since CLOCK works with only one event at a time, NEWT and NEWJ will always contain the time and type number of the present event being worked with. Each event type is considered to be the parameter of a computer GO TO statement in the user's program and represents the jump address of an event to be executed (Ref. 1, Appendix).

The variable L takes on integer values from 2 through 9; the value must be chosen by the user and determines the function of the call statement. *Example:* Consider the use of L in the following calls to the CLOCK subroutine.

L = 2: used for loading the Master Time List with the event (NEWT, NEWJ)—

continued

```
      CALL CLOCK(2,NEWT,NEWJ)
```

The parameter value of L equal to 2 denotes that the Master Time List is to be loaded with event type NEWJ that is scheduled to occur at time NEWT.

L = 3: used to delete from both lists all events of a given type NEWJ occurring at a given time NEWT—

```
      CALL CLOCK(3,NEWT,NEWJ)
```

This call to CLOCK is used to delete the occurrence of event type NEWJ at a scheduled time of NEWT from both the Master Time List and the Master Time Queue. For example, CALL CLOCK(3,8,1) requests that event (8,1) be deleted from both lists.

L = 4: used for calling the next scheduled event—

```
      CALL CLOCK(4,NEWT,NEWJ)
```

Here, a request has been made of the CLOCK for the next scheduled event and its time of occurrence. CLOCK returns a value in NEWT equal to the time of occurrence of the next event type. The value of the event-type code is returned in NEWJ.

L = 5: used for loading a blocked event (NEWT,NEWJ) on the Master Time Queue—

```
      CALL CLOCK(5,NEWT,NEWJ)
```

A blocked event (NEWT,NEWJ) is to be loaded onto the Master Time Queue. The user is cautioned that NEWT must *not* contain the time at which the event was blocked. If this is included, the clock will hang in an endless loop. NEWT is the *next* time of occurrence of the event type NEWJ *after* it has been blocked.

L = 6: produces the next scheduled time of occurrence (i.e., NEWT for an event of type NEWJ—

```
      CALL CLOCK(6,NEWT,NEWJ)
```

This call to the CLOCK requests the next scheduled time of occurrence of event type NEWJ. The answer to the request is returned in NEWT. For example, CALL CLOCK(6,NEWT,4) will return to the user (in NEWT_ the next time of occurrence of event type 4. If a request for the next scheduled time of occurrence of event type NEWJ cannot be satisfied (e.g., if that event type is not on the CLOCK lists), then the value returned in NEWT is negative.

continued

L = 7: purges (deletes) both lists of all events of type NEWJ—

```
      CALL CLOCK(7,0,NEWJ)
```

Here, the CLOCK is requested to delete from both the Master Time List and the Master Time Queue all occurrences of the event type specified by NEWJ. Note that the second parameter (NEWT) is denoted as 0, as this parameter is ignored by this request but must be filled in the calling sequence.

L 8: purges the Master Time List of all events of type NEWJ—

```
      CALL CLOCK(8,0,NEWJ)
```

This request is similar to 7, except that it deletes all events of type NEWJ from the Master Time List only.

L = 9: purges the Master Time Queue of all events of type NEWJ—

```
      CALL CLOSK(9,0,NEWJ)
```

This request is similar to 7 also, except that it deletes all events of type NEWJ from the Master Time Queue only.

NOTE: If an event is blocked for some reason, the user must place it on the Master Time Queue, at a later scheduled time, by using a different form of the CALL CLOCK (Ref. 1, p. 3.4).

The unit in which event times are measured is decided by the user in terms of the process being simulated. However, *event times must be nonnegative integers.*

The user has access to the values of NEVEN, NEVEQ, and LUN, used by the CLOCK. These values are placed in the labeled common block CLOCK1. (See Ref. 1, p.3.9.) NEVEN and NEVEQ should be accessed only on a read-only basis by the user. CLOCK sets and updates both variables.

If the user wants to run a real-time simulation, he can use FUNCTION TIMELP, which provides the elapsed CP time between successive calls.

TIMELP

```
REAL FUNCTION TIMELP (NPRINT)
TYPE INTEGER NPRINT
```

If the user wants to attempt a simulation in real time, he may find FUNCTION TIMELP useful. This routine prints (at the user's option) and returns the CDC 6400 central processor elapsed time (in seconds)

continued

between consecutive calls to TIMELP. An initializing call to TIMELP should be made at the beginning of execution of the routine to be timed. TIMELP can also be used to time any particular portion of a FORTRAN program.

A zero value of NPRINT suppresses printing by TIMELP. A nonzero NPRINT will print the legend CP TIME SINCE LAST CALL .. xx.xxx SECONDS. In either case, TIMELP contains the elapsed time. The call to TIMELP takes the form CALL TIMELP (NPRINT).

Job Deck Structure

```
Job Cards
Run(s)
LIBRARY(SPURT1,SPURT6)
LGO.
7-8-9 (EOR Card)
Main-Problem Card(s)
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
   .
   .
   .
   SUBROUTINE ACTIONZ
   .
   .
   .
   END
7-8-9 (EOR Card)
Data Cards
6-7-8-9 (EOI Card)
```

NOTE: The file assignments on the Main-Program header card are the standard SPURT assignments.

SAMPLE INPUT

```
                SUBROUTINE EXAMPL3                                                  SPTS9   0
          C                                                                         SPTS9  10
          C                                                                         SPTS9  20
          C     THIS PROGRAM IS PART OF THE NORTHWESTERN UNIVERSITY                 SPTS9  30
          C                                                                         SPTS9  40
          C                         S  P  U  R  T                                   SPTS9  50
          C                                                                         SPTS9  60
          C        A SIMULATION PACKAGE FOR UNIVERSITY RESEARCH AND TEACHING        SPTS9  70
          C                                                                         SPTS9  80
          C     ****************************************************************SPTS9  90
          C                                                                         SPTS9100
000002          COMMON /CLOCK1/ DUMM1,NEVEN,DUMM2,NEVEQ,LUN                         SPTS9110
000002          COMMON    MTL(500), JUMP(500), MTQ(100), JUMQ(100)                  SPTS9120
          C                                                                         SPTS9130
          C                                                                         SPTS9140
          C     THE FOLLOWING CARDS ARE COPIES OF THE EXOGENOUS EVENTS TO BE READ  SPTS9150
          C     NEWT    NEWJ                                                        SPTS9160
          C      2       5                                                          SPTS9170
          C      5       1                                                          SPTS9180
          C     24       8                                                          SPTS9190
          C                                                                         SPTS9200
          C     END OF EXOGENOUS EVENTS LIST                                        SPTS9210
          C                                                                         SPTS9220
          C     INITIALIZATION                                                      SPTS9230
000002          LMTL=500                                                            SPTS9240
000003          LMTQ=100                                                            SPTS9245
000004          L=1                                                                 SPTS9250
000005          MM=JJ=-1                                                            SPTS9260
000007          LUN=5                                                               SPTS9270
000010          DUM=TIMELP(0)                                                       SPTS9280
000013          CALL SETCLK (MTL,JUMP,LMTL,MTQ,JUMQ,LMTQ)                           SPTS9290
000017          DUM=TIMELP(1)                                                       SPTS9300
000022          PRINT 140                                                           SPTS9310
000025          PRINT 130,       MM,JJ,L,NEVEN,NEVEQ                                SPTS9320
          C                                                                         SPTS9330
          C     DUMMY ROUTINE TO LOAD ENDOGENOUS EVENTS ON MASTER_TIME_LIST         SPTS9340
          C                                                                         SPTS9350
000043          L=2                                                                 SPTS9360
000044          DO 20 I=1,10                                                        SPTS9370
000046          F=I                                                                 SPTS9380
000047          JJ=(ABS(COS(F)))*7.0+1.0                                            SPTS9390
000056          MM=2*I/JJ                                                           SPTS9400
000063          CALL CLOCK (L,MM,JJ)                                                SPTS9410
000065    20    PRINT 130,       MM,JJ,L,NEVEN,NEVEQ                                SPTS9420
000105          DUM=TIMELP(1)                                                       SPTS9430
          C                                                                         SPTS9440
          C     CALL OUT NEXT EVENT                                                 SPTS9450
          C                                                                         SPTS9460
000110    30    L=4                                                                 SPTS9470
000111          CALL CLOCK (L,MM,JJ)                                                SPTS9480
000114          PRINT 130,       MM,JJ,L,NEVEN,NEVEQ                                SPTS9490
000132          DUM=TIMELP(1)                                                       SPTS9500
000135          GO TO (100,200,300,400,500,600,700,800),JJ                          SPTS9510
          C                                                                         SPTS9520
000151    100   K=100                                                               SPTS9530
000152          PRINT 120, K,MM                                                     SPTS9540
000162          GO TO 30                                                            SPTS9550
```

continued

```
          C                                                                SPTS9560
000163    200    K=200                                                     SPTS9570
                 PRINT 120, K,MM                                           SPTS9580
000164
000174           L=5                                                       SPTS9590
000175           MM=MM+3                                                   SPTS9600
000177           JJ=3                                                      SPTS9610
000200           CALL CLOCK (L,MM,JJ)                                      SPTS9620
000203           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9630
000221           DUM=TIMELP(1)                                             SPTS9640
000224           GO TO 30                                                  SPTS9650
          C                                                                SPTS9660
000224    300    K=300                                                     SPTS9670
000225           PRINT 120, K,MM                                           SPTS9680
000235           GO TO 30                                                  SPTS9690
          C                                                                SPTS9700
000236    400    K=400                                                     SPTS9710
000237           PRINT 120, K,MM                                           SPTS9720
000247           L=3                                                       SPTS9730
000250           J=NEVEN/2+1                                               SPTS9740
000252           JJ=JUMP(J)                                                SPTS9750
000255           MM=MTL(J)                                                 SPTS9760
000260           CALL CLOCK (L,MM,JJ)                                      SPTS9770
000262           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9780
000300           DUM=TIMELP(1)                                             SPTS9790
000303           GO TO 30                                                  SPTS9800
          C                                                                SPTS9810
000303    500    K=500                                                     SPTS9820
000304           PRINT 120, K,MM                                           SPTS9830
000314           L=6                                                       SPTS9840
000315           JJ=3                                                      SPTS9850
000316           CALL CLOCK (L,MM,JJ)                                      SPTS9860
000321           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9870
000337           DUM=TIMELP(1)                                             SPTS9880
000341           L=9                                                       SPTS9890
000343           CALL CLOCK (L,MM,JJ)                                      SPTS9900
000345           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9910
000363           DUM=TIMELP(1)                                             SPTS9920
000366           GO TO 30                                                  SPTS9930
          C                                                                SPTS9940
000366    600    K=600                                                     SPTS9950
000367           PRINT 120, K,MM                                           SPTS9960
000377           L=2                                                       SPTS9970
000400           MM=MM+5                                                   SPTS9980
000402           JJ=1                                                      SPTS9990
000403           CALL CLOCK (L,MM,JJ)                                      SPTS9   0
000406           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9  10
000424           DUM=TIMELP(1)                                             SPTS9  20
000427           GO TO 30                                                  SPTS9  30
          C                                                                SPTS9  40
000427    700    K=700                                                     SPTS9  50
000430           PRINT 120, K,MM                                           SPTS9  60
000440           L=8                                                       SPTS9  70
000441           CALL CLOCK (L,MM,JJ)                                      SPTS9  80
000444           PRINT 130,        MM,JJ,L,NEVEN,NEVEQ                     SPTS9  90
000462           DUM=TIMELP(1)                                             SPTS9100
000465           GO TO 30                                                  SPTS9110
          C                                                                SPTS9120
000465    800    K=800                                                     SPTS9130
```

continued

```
000466         PRINT 120, K,MM                                                SPTS9140
000476         L=7                                                            SPTS9150
000477         CALL CLOCK (L,MM,JJ)                                           SPTS9160
000502         PRINT 130,       MM,JJ,L,NEVEN,NEVEQ                           SPTS9170
000520         DUM=TIMELP(1)                                                  SPTS9180
000523         GO TO 30                                                       SPTS9190
          C                                                                   SPTS9200
          C                                                                   SPTS9210
000523    120  FORMAT (1H0,*ENTITY *,I5,*  _ACTIVATED AT SIMULATION TIME =  *,I8)SPTS9220
000523    130  FORMAT (1H0,30X           I8,14X,I8,14X,I8,10X,I10,10X,I10)    SPTS9230
000523      140 FORMAT(30X,     *SIMULATED TIME*10X*JUMP ADDR*19X,            SPTS9240
             1 *L*16X*NEVEN*16X*NEVEQ*///)                                    SPTS9250
000523         END                                                            SPTS9260
```

000 0071
SPURT1

SAMPLE OUTPUT

```
*****     CLOCK        *****         DATE  01/27/70      PAGE NO.  49        NORTHWESTERN UNIVERSITY  S P U R T

  .. CP TIME SINCE LAST CALL ..    .047 SECONDS
                               SIMULATED TIME         JUMP ADDR              L                NEVEN             NEVEQ


                                      -1                 -1                  1                  0                0.
                                       0                  4                  2                  1                0.
                                       1                  3                  2                  2                0.
                                       0                  7                  2                  3                0.
                                       1                  5                  2                  4                0.
                                       5                  2                  2                  5                0.
                                       1                  7                  2                  6                0.
                                       2                  6                  2                  7                0.
                                       8                  2                  2                  8                0.
                                       2                  7                  2                  9                0.
                                       3                  6                  2                 10                0.
  .. CP TIME SINCE LAST CALL ..    .046 SECONDS

                                       0                  4                  4                  9                0.
  .. CP TIME SINCE LAST CALL ..    .006 SECONDS

ENTITY   400   ACTIVATED AT SIMULATION TIME =       0.
                                       2                  6                  3                  8                0.
  .. CP TIME SINCE LAST CALL ..    .009 SECONDS
                                       .                  .                  .
                                       .                  .                  .
                                       .                  .                  .
```

continued

```
ERROR ENCOUNTERED IN ROUTINE ...CLOCK

     REQUEST FOR NEXT SCHEDULED TIME OF OCCURANCE OF EVENT UNSATISFIED ... EVENT NOT ON LISTS

PROGRAM CONTINUING...

****************************************************************************************************
                  -99999               3                   6                  3                   0
 .. CP TIME SINCE LAST CALL ..    .039 SECONDS
                  -99999               0                   9                  3                   0
 .. CP TIME SINCE LAST CALL ..    .006 SECONDS
                       3               6                   4                  2                   0
 .. CP TIME SINCE LAST CALL ..    .006 SECONDS
                                       .                   .                  .
                                       .                   .                  .
                                       .                   .                  .

ERROR ENCOUNTERED IN ROUTINE ...CLOCK

     BOTH CLOCK LISTS EXHAUSTED.

JOB TERMINATED.
```

USER INSTRUCTIONS—SPURT2: Stochastic Generators

The STOGN (stochastic-generating) functions enable the user to generate samples from various user-defined probability distributions and calculate sample values.

All of the STOGN functions that produce random values use the standard library random-number-generating function RANF. Unless otherwise preset, the seed (and, therefore, the entire stream of numbers generated) is the same for each run. To change the seed, the user should CALL RANSET (SEED) during his simulation system initialization. A convenient value for SEED may be determined by SEED = TIME(0). This should produce a new seed each time.

NOTE: The user is cautioned to use a TYPE statement in all of his routines that use the stochastic generators. Failure to specify the routine name as a TYPE INTEGER could result in improper evaluation of mixed-mode expressions.

STOGN1

```
REAL FUNCTION STOGN1(NN,XXX,ORD,K)
TYPE REAL XXX, ORD
TYPE INTEGER NN, K
```

STOGN1 is a discrete, empirical, probability distribution, defined by the user. This FORTRAN function accesses a user-defined probability distribution that is called by VALUE = STOGN1 (NN,XXX,ORD,K). For each empirical probability distribution that the user employs, he must define two two-dimensional arrays XXX and ORD, with the 2nd subscript of these two arrays being the identification number that the user assigns to the probability distribution. K is the name of the probability distribution. Any allowable variable names of TYPE REAL may be substituted for XXX and ORD. The arrays must be dimensioned in the user's program, along with defining the variable NN.

STOGN2

```
REAL FUNCTION STOGN2(NN,XXX,ORD,K)
TYPE REAL XXX, ORD
TYPE INTEGER NN, K
```

STOGN2 enables the user to approximate a continuous probability distribution by means of a piecewise linear one. The statement VALUE = STOGN2(NN,XXX,ORD,K) will cause to be returned values that represent a random sample from a particular distribution. Otherwise, the same rules apply to the general case for STOGN2 as for STOGN1.

continued

000 0071
SPURT2

STOGN6

REAL FUNCTION STOGN6(N,P)
TYPE REAL P
TYPE INTEGER N

STOGN6 enables th user to obtain a random sample from the negative binomial distribution, with parameters N (number of successful trials) and P (probability of success). S is the value returned by STOGN6 and represents the number of unsuccessful trials. The call is of the form S = STOGN6(N,P). The user is cautioned to observe the restrictions of $N \geq 0$ and $0.0 < P < 1.0$.

UNIFRM, NORMAL

REAL FUNCTION UNIFRM(A,B)
REAL FUNCTION NORMAL(A,B)
TYPE REAL A,B

UNIFRM obtains a sample of real values that are equally likely to fall between the values A and B, where $A < B$. The user is able to sample from a uniform distribution defined on the interval from A to B. The function is called by XNUM = UNIFRM(A,B).

NORMAL allows the user to obtain a random sample from a normal distribution, with mean A and standard deviation B. FUNCTION NORMAL is called by XNUM = NORMAL(A,B).

RANDIN

INTEGER FUNCTION RANDIN(A,B)
TYPE INTEGER A,B

RANDIN(A,B) provides a uniform distribution of integers between the limits of A and B. The value is one of the integers A, A + 1, ···, B – 1, B, where $A < B$, with equal probability of each. The call to this function is of the form NUM = RANDIN(A,B).

LINEAR, DRAW

REAL FUNCTION LINEAR(A,B)
LOGICAL FUNCTION DRAW(A)
TYPE REAL A,B

LINEAR provides the user with a random supply from a (cumulative) distribution function F, which is obtained by linear interpolation in a nonequidistant table defined by A and B, such that A(i) = F(B(i)). It is assumed that A and B are one-dimensional real arrays of the same length, that the first and last elements of A are equal to 0.0

continued

and 1.0, respectively, and that A(i) < A(j) and B(i) < B(j) for i < j. The call to LINEAR is XNUM = LINEAR(A,B).

DRAW returns a value of .TRUE. with the probability A and .FALSE. with the probability 1 − A. DRAW is always .TRUE. if A ≥ 1 and is always .FALSE. if A ≤ 0. Consider the following example of the use of the function DRAW: IF (DRAW(0.72)) 20, 30. This will transfer control to statement 20 with a probability of occurrance of 0.72 and to statement 30 with a probability of 0.28.

NEGEXP, ERLANG

```
REAL FUNCTION NEGEXP(A)
REAL FUNCTION ERLANG(A,B)
TYPE REAL A,B
```

NEGEXP obtains a random sample from the negative exponential distribution with mean 1/A, defined by −(1/A) [ln(random number)]. This is the same as random "waiting time" in a Poisson-distributed arrival pattern with expected number of arrivals per unit time equal to A. A call to FUNCTION NEGEXP could be XNUM = NEGEXP(A) or XNUM = NEGEXP(1.0/XMEAN).

ERLANG obtains a random sample from the Erlang distribution with mean 1/A and standard deviation 1/(A√B). It is defined by B basic drawings of a random number, if B is an integer value, or, otherwise, by B + 1 drawings. Both A and B must be greater than zero in the call to ERLANG: XNUM = ERLANG(A,B).

POISSN

```
INTEGER FUNCTION POISSN(A)
TYPE REAL A
```

POISSN provides a random sample from the Poisson distribution with parameter A. It is obtained by n + 1 basic drawings of random numbers, where n is the function value defined as the smallest nonnegative integer for which

$$\prod_{i=0}^{n} \mathrm{RANF}(0) < e^{-A},$$

where RANF is the random-number-generating function. When the parameter of A is greater than 20.0, the value is approximated by the integer value of NORMAL(A, SQRT(A)) or, when this is negative, by zero. A sample call is of the form NUM = POISSN(A).

continued

000 0071
SPURT2

DISCRT

INTEGER FUNCTION DISCRT(A)
TYPE REAL A

The one-dimensional real array A is interpreted as a step function of the subscript, defining a discrete (cumulative) distribution function. The function value is an integer in the range [lsb – 1, usb – 1], where lsb and usb are the lower and upper subscript bounds of the array. It is defined as the smallest i, such that $A(i + 1) > \mu$, where μ is the basic random number produced by RANF and where A(usb) = 1.0. The call is of the form NUM = DISCRT(A).

HISTD

INTEGER FUNCTION HISTD(A)
TYPE REAL A

HISTD provides the user with the random value of an integer in the range [lsb, usb], where lsb and usb are the lower and upper subscript bounds of the one-dimensional real array A. The latter is interpreted as a histogram defining the relative frequencies of the values.

This function is more time-consuming than the function DISCRT, where the cumulative distribution function is given, but it is more useful if the frequency histogram is dynamically updated at run time. A sample call is NUM = HISTD(A).

RANPER

SUBROUTINE RANPER(N,M)
TYPE INTEGER N,M

Subroutine RANPER generates a uniformly distributed, random permutation of the integers 1, 2, ···, M and stores it in the array N, where N is an integer-valued array of dimension $M(\geq 1)$. RANPER is called by CALL RANPER(N,M).

continued

Job Deck Structure

```
Job Cards
Run(s)
LIBRARY(SPURT1,SPURT6)
LGO.
7–8–9 (EOR Card)
Main-Problem Cards
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
   .
   .
   .
   SUBROUTINE ACTIONZ
   .
   .
   .
   END
7–8–9 (EOR Card)
Data Cards
6–7–8–9 (EOI Card)
```

NOTE: The file assignments shown on the Main-Program header card are the standard SPURT assignments.

SAMPLE INPUT

```
                  SUBROUTINE TESTGEN                                              SPTS1   0
             C                                                                    SPTS1  10
             C    ******************************************************************SPTS1  20
             C                                                                    SPTS1  30
             C    THIS PROGRAM IS PART OF THE NORTHWESTERN UNIVERSITY              SPTS1  40
             C                                                                    SPTS1  50
             C                        S  P  U  R  T                               SPTS1  60
             C                                                                    SPTS1  70
             C       A SIMULATION PACKAGE FOR UNIVERSITY RESEARCH AND TEACHING     SPTS1  80
             C                                                                    SPTS1  90
             C    ******************************************************************SPTS1100
             C                                                                    SPTS1110
             C                                                                    SPTS1120
             C    ******************************************************************SPTS1130
             C                                                                    SPTS1140
             C    THIS IS A TEST PROGRAM TO CHECK THE RESULTS OF THE               SPTS1150
             C    THE RANDOM NUMBER GENERATORS OF THE SPURT PACKAGE                SPTS1160
             C    THIS ROUTINE WILL ALSO CHECK THE PERFORMANCE OF THE HISTOGRAM    SPTS1170
             C    AND PLOT PACKAGE  ( STIX1,STIX2,STIX3,GRAPH,ENDPLT)              SPTS1180
             C                                                                    SPTS1190
             C    ******************************************************************SPTS1200
             C                                                                    SPTS1210
000002            INTEGER RANDIN,POISSN,HISTD,DISCRT                              SPTS1220
000002            TYPE LOGICAL DRAW                                               SPTS1230
000002            REAL NORMAL,LINEAR                                              SPTS1240
000002            REAL NEGEXP                                                     SPTS1250
000002            DIMENSION X4(5)                                                 SPTS1260
000002            COMMON /QPLOTQ/ A(5),PLOTSW                                     SPTS1270
000002            DIMENSION XXX2(3,2), ORD2(3,2), XXX1(5,1), ORD1(5,1)            SPTS1280
000002            COMMON VALUES1(104)                                             SPTS1290
                 1         ,VALUES2(104), VALUES3(104), VALUES7(104), VALUES8(104), SPTS1300
                 1VALUES9(104)                                                    SPTS1310
000002            DATA (X4=.15,.35,.35,1.5,1.0)                                   SPTS1320
000002            DATA (XXX2=0.2,0.7,1.0,0.4,0.99,1.0),(ORD2=4.0,5.0,6.0,11.0,15.0,1SPTS1330
                 15.0),(XXX1=0.0,0.15,0.50,0.85,1.00),(ORD1=0.0,1.0,2.0,3.0,4.0)   SPTS1340
000002            DUM=TIMELP(0)                                                   SPTS1350
000005            CALL STIX1(VALUES2,7.0,3.0,40.,VALUE,6HSTOGN1,6)                SPTS1360
000013            DO 10 I=1,10000                                                 SPTS1370
000015            VALUE=STOGN1(3,XXX2,ORD2,1)                                     SPTS1380
000021            CALL STIX2(VALUES2,7.0,3.0,40.,VALUE,6HSTOGN1,6)                SPTS1390
000027         10 CONTINUE                                                        SPTS1400
000031            CALL STIX3(VALUES2,7.0,3.0,40.,VALUE,6HSTOGN1,6)                SPTS1410
000040            DUM=TIMELP(1)                                                   SPTS1420
000043            CALL STIX1(VALUES1,4.0,0.0,40.,VALUE,6HSTOGN2,6)                SPTS1430
000051            DO 30 I=1,10000                                                 SPTS1440
000053            VALUE=STOGN2(5,XXX1,ORD1,1)                                     SPTS1450
000057            CALL STIX2(VALUES1,4.0,0.0,40.,VALUE,6HSTOGN2,6)                SPTS1460
0Q0065         30 CONTINUE                                                        SPTS1470
000067            CALL STIX3(VALUES1,4.0,0.0,40.,VALUE,6HSTOGN2,6)                SPTS1480
000076            DUM=TIMELP(1)                                                   SPTS1490
000101            CALL STIX1(VALUES9,40.0,0.0,40.,VALUE,6HSTOGN6,6)               SPTS1500
000107            DO 70 I=1,10000                                                 SPTS1510
000111            VALUE=STOGN6(10,0.5)                                            SPTS1520
000114            CALL STIX2(VALUES9,40.0,0.0,40.,VALUE,6HSTOGN6,6)               SPTS1530
000122   70       CONTINUE                                                        SPTS1540
000124            CALL STIX3(VALUES9,40.0,0.0,40.,VALUE,6HSTOGN6,6)               SPTS1550
000133            DUM=TIMELP(1)                                                   SPTS1560
```

continued

```
000136          CALL STIX1(VALUES1,21.0,-21.0,42,VALUE,6HRANDIN,6)              SPTS1570
000144          DO 90 I=1,10000                                                 SPTS1580
000146          VALUE=RANDIN(-20,20)                                            SPTS1590
000151          CALL STIX2(VALUES1,21.0,-21.0,42,VALUE,6HRANDIN,6)              SPTS1600
000160    90    CONTINUE                                                        SPTS1610
000162          CALL STIX3(VALUES1,21.0,-21.0,42,VALUE,6HRANDIN,6)              SPTS1620
000171          DUM=TIMELP(1)                                                   SPTS1630
000174          CALL STIX1(VALUES1,4.0,-4.0,40,VALUE,6HNORMAL,6)                SPTS1640
000202          DO 100 I=1,10000                                                SPTS1650
000204          VALUE=NORMAL(0.0,1.0)                                           SPTS1660
000207          CALL STIX2(VALUES1,4.0,-4.0,40,VALUE,6HNORMAL,6)                SPTS1670
000215   100    CONTINUE                                                        SPTS1680
000217          CALL STIX3(VALUES1,4.0,-4.0,40,VALUE,6HNORMAL,6)                SPTS1690
000226          DUM=TIMELP(1)                                                   SPTS1700
000231          CALL STIX1(VALUES1,20.0,0.0,40,VALUE,6HPOISSN,6)                SPTS1710
000237          DO 110 I=1,10000                                                SPTS1720
000241          VALUE=POISSN(5.0)                                               SPTS1730
000243          CALL STIX2(VALUES1,20.0,0.0,40,VALUE,6HPOISSN,6)                SPTS1740
000252   110    CONTINUE                                                        SPTS1750
000254          CALL STIX3(VALUES1,20.0,0.0,40,VALUE,6HPOISSN,6)                SPTS1760
000263          DUM=TIMELP(1)                                                   SPTS1770
000266          CALL STIX1(VALUES1,10.0,0.0,40,VALUE,6HERLANG,6)                SPTS1780
000274          DO 130 I=1,10000                                                SPTS1790
000276          VALUE=ERLANG(0.5,2.0)                                           SPTS1800
000301          CALL STIX2(VALUES1,10.0,0.0,40,VALUE,6HERLANG,6)                SPTS1810
000307   130    CONTINUE                                                        SPTS1820
000311          CALL STIX3(VALUES1,10.0,0.0,40,VALUE,6HERLANG,6)                SPTS1830
000320          DUM=TIMELP(1)                                                   SPTS1840
000323          CALL STIX1(VALUES1,4.5,.5,40,VALUE,5HHISTD,5)                   SPTS1850
000331          DO 140 I=1,10000                                                SPTS1860
000333          VALUE=HISTD(X4)                                                 SPTS1870
000335          CALL STIX2(VALUES1,4.5,.5,40,VALUE,5HHISTD,5)                   SPTS1880
000344   140    CONTINUE                                                        SPTS1890
000346          CALL STIX3(VALUES1,4.5,.5,40,VALUE,5HHISTD,5)                   SPTS1900
000355          DUM=TIMELP(1)                                                   SPTS1910
000360          CALL STIX1(VALUES1,1.5,-.5,10,VALUE,4HDRAW,4)                   SPTS1990
000366          DO 170 I=1,10000                                                SPTS1   0
000370          IF(DRAW(0.69)) 162,164                                          SPTS1  10
000373      162 VALUE=1.0                                                       SPTS1  20
000375          GO TO 166                                                       SPTS1  30
000375      164 VALUE=0.0                                                       SPTS1  40
000376      166 CONTINUE                                                        SPTS1  50
000376          CALL STIX2(VALUES1,1.5,-.5,10,VALUE,4HDRAW,4)                   SPTS1  60
000405   170    CONTINUE                                                        SPTS1  70
000407          CALL STIX3(VALUES1,1.5,-.5,10,VALUE,4HDRAW,4)                   SPTS1  80
000416          DUM=TIMELP(1)                                                   SPTS1  90
000421          RETURN                                                          SPTS1100
000421          END                                                             SPTS1110
```

SAMPLE OUTPUT

```
*****    STIX3        *****          DATE  01/27/70     PAGE NO.   2         NORTHWESTERN UNIVERSITY  S P U R T

                                THIS DATA IS ACCOMPANIED BY GRAPH NUMBER          1

               STOGN1

               ENTRIES IN TABLE     MEAN ARGUMENT     STANDARD DEVIATION
                          10000             5.101                   .700

               UPPER          OBSERVED        PER CENT      CUMULATIVE      CUMULATIVE       MULTIPLE       DEVIATION      CUMULATIVE
               LIMIT          FREQUENCY       OF TOTAL      PERCENTAGE       REMAINDER        OF MEAN       FROM MEAN     PROBABILITY
 UNDERFLOW
               3.000                  0          0.000           0.000         100.000           .588          -3.001            .001
               3.100                  0          0.000           0.000         100.000           .608          -2.858            .002
               3.200                  0          0.000           0.000         100.000           .627          -2.715            .003
               3.300                  0          0.000           0.000         100.000           .647          -2.573            .005
               3.400                  0          0.000           0.000         100.000           .667          -2.430            .008
                                           .                  .                            .                  .
                                           .                  .                            .                  .
                                           .                  .                            .                  .

*****    GRAPH        *****          DATE  01/27/70     PAGE NO.   3         NORTHWESTERN UNIVERSITY  S P U R T


               GRAPH NUMBER     1 HAS JUST BEEN PLOTTED ON CALCOMP

         THE PLOT IS OF ...

         STOGN1

                 VS.

                    FREQUENCY (PER CENT)


         END OF OUTPUT BY GRAPH


            .. CP TIME SINCE LAST CALL ..   5.395 SECONDS
```

continued

000 0071
SPURT2

```
*****     STIX3         *****         DATE  01/27/70      PAGE NO.  10         NORTHWESTERN UNIVERSITY  S P U R T

                                 THIS DATA IS ACCOMPANIED BY GRAPH NUMBER          5

                NORMAL

                ENTRIES IN TABLE     MEAN ARGUMENT     STANDARD DEVIATION
                      10000               .007                  1.006

                  UPPER        OBSERVED        PER CENT      CUMULATIVE      CUMULATIVE         MULTIPLE      DEVIATION     CUMULATIVE
                  LIMIT        FREQUENCY       OF TOTAL      PERCENTAGE      REMAINDER           OF MEAN      FROM MEAN     PROBABILITY
UNDERFLOW
                 -4.000               0          0.000           0.000         100.000          -567.722        -3.984            .000
                 -3.800               0          0.000           0.000         100.000          -539.336        -3.785            .000
                 -3.600               1           .010            .010          99.990          -510.950        -3.586            .000
                 -3.400               0          0.000            .010          99.990          -482.564        -3.387            .000
                 -3.200               8           .080            .090          99.910          -454.178        -3.189            .001
                 -3.000              11           .110            .200          99.800          -425.791        -2.990            .001
                 -2.800              15           .150            .350          99.650          -397.405        -2.791            .003
                             .               .               .                               .             .              .
                             .               .               .                               .             .              .
                             .               .               .                               .             .              .
                  3.000              18           .180          99.860            .140           425.791         2.976            .999
                  3.200               5           .050          99.910            .090           454.178         3.175            .999
                  3.400               6           .060          99.970            .030           482.564         3.373           1.000
                  3.600               2           .020          99.990            .010           510.950         3.572           1.000
                  3.800               0          0.000          99.990            .010           539.336         3.771           1.000
OVERFLOW  XXXXXXXXXX4.000             1           .010         100.000            .000           567.722         3.970           1.000
                  4.200               0          0.000         100.000            .000           596.108         4.169           1.000
```

000 0071

000 0071

continued

```
*****     STIX3          *****          DATE  01/27/70      PAGE NO.  18        NORTHWESTERN UNIVERSITY  S P U R T

                                        THIS DATA IS ACCOMPANIED BY GRAPH NUMBER          9

              DRAW

              ENTRIES IN TABLE      MEAN ARGUMENT       STANDARD DEVIATION
                     10000                  .690                    .463

              UPPER         OBSERVED          PER CENT      CUMULATIVE      CUMULATIVE        MULTIPLE        DEVIATION      CUMULATIV
              LIMIT         FREQUENCY         OF TOTAL      PERCENTAGE      REMAINDER          OF MEAN        FROM MEAN      PROBABILIT
UNDERFLOW
              -.500                 0            0.000           0.000        100.000            -.725           -2.572            .00
              -.300                 0            0.000           0.000        100.000            -.435           -2.139            .01
              -.100                 0            0.000           0.000        100.000            -.145           -1.707            .04
               .100              3103           31.030          31.030         68.970             .145           -1.275            .10
               .300                 0            0.000          31.030         68.970             .435            -.842            .20
               .500                 0            0.000          31.030         68.970             .725            -.410            .34
               .700                 0            0.000          31.030         68.970            1.015             .022            .50
               .900                 0            0.000          31.030         68.970            1.305             .455            .67
              1.100              6897           68.970         100.000           .000            1.595             .887            .81
              1.300                 0            0.000         100.000           .000            1.885            1.319            .90
OVERFLOW  XXXXXXXX1.500             0            0.000         100.000           .000            2.175            1.752            .96
              1.700                 0            0.000         100.000           .000            2.465            2.184            .98
```

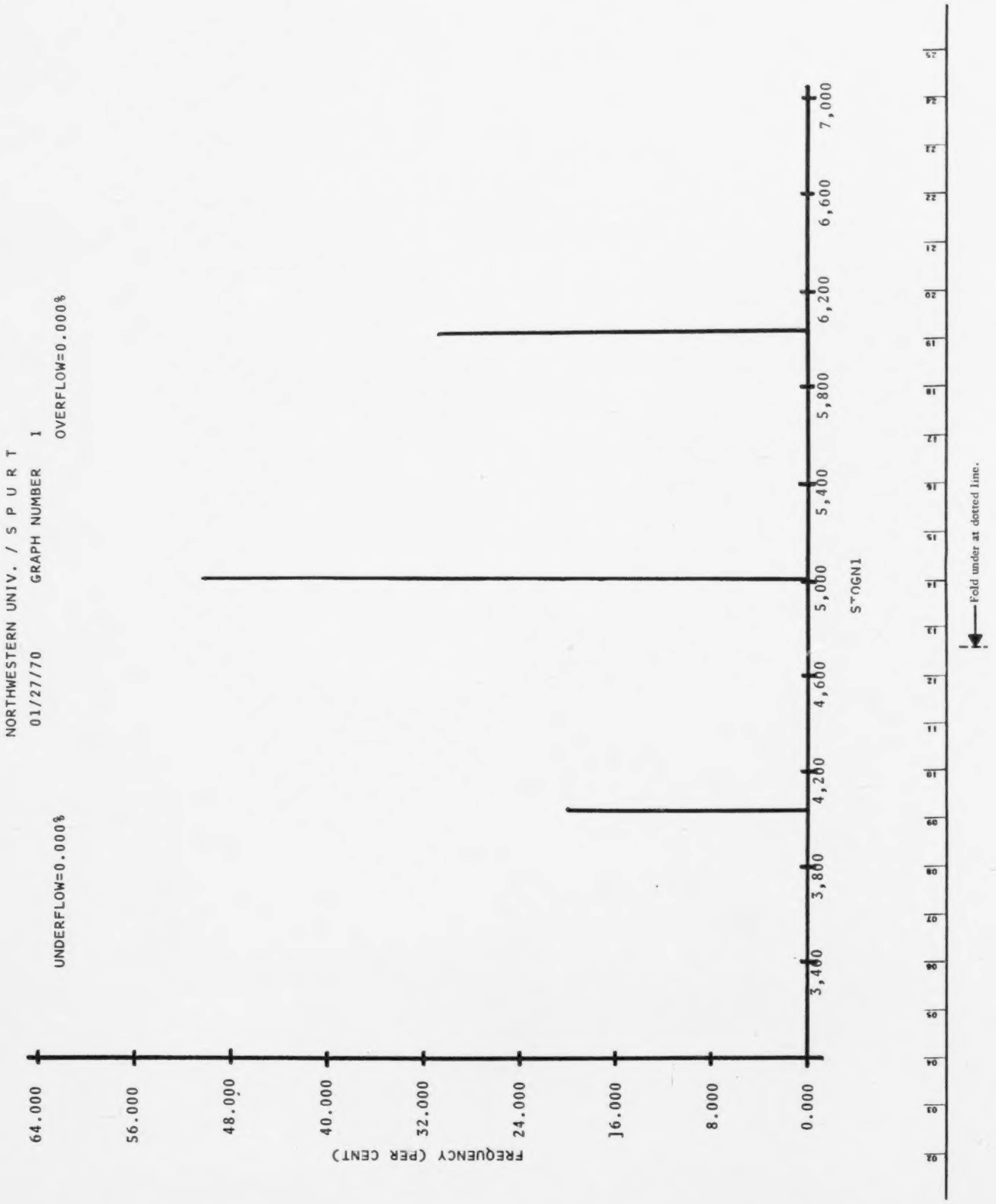
```
*****       INSCUT        *****           DATE  01/27/70      PAGE NO.  18         NORTHWESTERN UNIVERSITY  S P U R T

       C1          C2          C3          C4          C5          C6          C7          C8          C9          C10
            201         202         203         204         205         206         207         208         209         210
```

continued

000 0071
SPURT2

```
*****     STIX3        *****        DATE  01/27/70      PAGE NO.  16        NORTHWESTERN UNIVERSITY  S P U R T

                                    THIS DATA IS ACCOMPANIED BY GRAPH NUMBER          8

               HISTD

               ENTRIES IN TABLE     MEAN ARGUMENT     STANDARD DEVIATION
                          10000             2.500                   .915

                 UPPER      OBSERVED       PER CENT    CUMULATIVE    CUMULATIVE      MULTIPLE    DEVIATION    CUMULATIVE
                 LIMIT     FREQUENCY       OF TOTAL    PERCENTAGE     REMAINDER       OF MEAN    FROM MEAN   PROBABILITY
 UNDERFLOW
                  .500             0          0.000         0.000       100.000          .200       -2.187          .014
                  .600             0          0.000         0.000       100.000          .240       -2.078          .019
                  .700             0          0.000         0.000       100.000          .280       -1.969          .025
                  .800             0          0.000         0.000       100.000          .320       -1.859          .032
                  .900             0          0.000         0.000       100.000          .360       -1.750          .040
                 1.000             0          0.000         0.000       100.000          .400       -1.640          .050
                 1.100          1471         14.710        14.710        85.290          .440       -1.531          .063
                         .               .                         .             .
                         .               .                         .             .
                         .               .                         .             .
                 4.000             0          0.000        85.390        14.610         1.600        1.640          .949
                 4.100          1461         14.610       100.000          .000         1.640        1.749          .960
                 4.200             0          0.000       100.000          .000         1.680        1.859          .968
                 4.300             0          0.000       100.000          .000         1.720        1.968          .975
                 4.400             0          0.000       100.000          .000         1.760        2.077          .981
 OVERFLOW  XXXXXXXXX4.500          0          0.000       100.000          .000         1.800        2.187          .986
                 4.600             0          0.000       100.000          .000         1.840        2.296          .989



*****     INSOUT       *****        DATE  01/27/70      PAGE NO.  32        NORTHWESTERN UNIVERSITY  S P U R T

      C1          C2          C3          C4          C5          C6          C7          C8          C9          C10

            11          12          13          14          15          16          17          18          19          20
```

continued


NORTHWESTERN UNIV. / S P U R T
01/27/70
GRAPH NUMBER 1
UNDERFLOW=0.000%
OVERFLOW=0.000%
64.000
56.000
48.000
40.000
32.000
24.000
16.000
8.000
0.000
FREQUENCY (PER CENT)
3,400
3,800
4,200
4,600
5,000
5,400
5,800
6,200
6,600
7,000
STOGN1
Fold under at dotted line.

USER INSTRUCTIONS—SPURT3: Statistical Computations

STIX1, STIX2, STIX3

```
SUBROUTINE STIX1, STIX2, and STIX 3 (TABLE,VMAX,VMIN,NINK,VAB,
   ITITLE,LNTHTL)
TYPE REAL VAB,TABLE,VMAX,VMIN
TYPE INTEGER NINK,LNTHL
ITITLE is a Hollerith constant
```

These three subroutines accumulate and print out a frequency table and, optionally, plot a CalComp histogram. The calls to these STIX are

```
CALL STIX1(TABLE,VMAX,VMIN,NINK,VAB,ITITLE,LNTHTL)
CALL STIX2(TABLE,VMAX,VMIN,NINK,VAB,ITITLE,LNTHTL)
CALL STIX3(TABLE,VMAX,VMIN,NINK,VAB,ITITLE,KNTHTL)
```

VAB is the variable being accumulated; TABLE is the frequency table used for accumulating a count of the different values of the variable VAB; NINK is the number of increments desired in the table; VMAX and VMIN are the maximum and minimum values, respectively, in the table.

The dimension TABLE must be specified in the user's program and *must have at least NINK + 4 entries.* The additional four entries count those values of VAB that are either greater or less than the range specified by VMAX and VMIN (i.e., overflow and underflow) and also accumulate a running sum and sum of squares for use by STIX3.

ITITLE is a Hollerith constant (defined either in a DATA statement or in the calling sequence) that is used to label the plot produced by STIX. LNTHTL is the length of the title, specified as the *number of characters in ITITLE.*

NOTE: The user is required to pass the same parameters to STIX1, STIX2, and STIX3 for each user-defined frequency table being built and plotted.

The three routines are used as follows. STIX1 will initialize the frequency table to zero. STIX2 will add a one to the proper counter in TABLE associated with the current value of VAB. STIX3 will print out the frequency table TABLE and a plot of the histogram will occur at the user's option. If a CalComp plot has been made, the user is cautioned to place a call to ENDPLT before any possible program terminations. (See Ref. 1, p. 6.3.)

continued

STIX4

```
SUBROUTINE STIX4 (X,N,XMIN,XMAX,STADEV,XBAR)
TYPE REAL X,XMIN,XMAX,STADEV,XBAR
TYPE INTEGER N
```

STIX4 takes a one-dimensional array X(1), ···, X(N), which represents values of a variable X, and computes the mean (XBAR), standard deviation (STADEV), maximum value (XMAX), and the minimum value (XMIN). The call is of the form CALL STIX4 (X,N,XMIN, XMAX,STADEV,XBAR), where X is an array of length N that is defined in the calling program. XMIN, XMAX, STADEV, and XBAR are the values returned by the arguments of STIX4.

STIX5

```
SUBROUTINE STIX5 (X,Y,N,CORR)
TYPE REAL X,Y,CORR
TYPE INTEGER N
```

STIX5 requires two one-dimensional arrays, X(1), ···, X(N) and Y(1), ···, Y(N), which represent the values of two variables X and Y. If the correlation coefficient of these two arrays is desired, then the following call is made to STIX5: CALL STIX5 (X, Y,N,CORR). This call will return to the user in CORR the value of the correlation coefficient between the two arrays X and Y.

STIX6

```
SUBROUTINE STIX6 (A,N,AMEDIAN,RANGE)
TYPE REAL A,AMEDIAN,RANGE
TYPE INTEGER N
```

STIX6 is designed to rank a one-dimensional array A and return the value of the median and the range of the data contained in the N data entries of A. STIX6 is called by CALL STIX6 (A,N, AMEDIAN,RANGE). The routine uses a bubble sort to rank the data. This enables the user to call STIX6, with dummy parameters for AMEDIAN and RANGE, and use it just to sort a one-dimensional array.

NOTE: For ranking two-dimensional arrays, see List-Processing and Queue-Manipulation, SPURT4: subroutines RANKQS, IRNKQS, RANKQL, and IRNKQL.

STIX7

```
SUBROUTINE STIX7 (X,N,XI,XF,DX,MUNIT,ISTAT,MODE,IHIST,ITITLE,LNTH)
TYPE REAL X,XI,XF,DX
```

continued

TYPE INTEGER N,ISTAT,MODE,IHIST,LNTH
MUNIT and ITITLE are Hollerith constants

The call for STIX7 is CALL STIX7 (X,N,XI,XF,DX,MUNIT,ISTAT,MODE, IHIST,ITITLE,LNTH). STIX7 will produce a statistical description of the data found in the first N locations of array X. MUNIT is a variable that contains the units in which the members of X are measured. ISTAT is a control parameter. If its value is nonzero, the sample size, mean, standard deviation, standard error, minimum value, maximum value, and range will be printed out. If ISTAT equals zero, these will be suppressed.

In a similar manner, a nonzero value for MODE will cause the mode to be printed out and a nonzero value of IHIST will cause a histogram to be printed on the line printer. ITITLE is the title of the histogram and consists of LNTH characters, where LNTH ≤ 90. If the mode and/or histogram is desired, XI, XF, and DX must be defined. (They must always be included in the calling sequence, however.) XI denotes the initial value for the histogram; XF is the final value; DX is the increment. STIX7 is set up to handle 100 increments between XI and XF. The value printed for each DX is the benchmark (median) value of that interval.

WARNING: If ISTAT = MODE = IHIST = 0, *no printout will result. Make sure that at least one of these parameters is nonzero.*

Job Deck Structure

```
Job Cards
Run(s)
LIBRARY(SPURT2,SPURT3,SPURT6)
LGO.
7-8-9 (EOR Card)
Main-Problem Card(s)
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
   .
   .
   .
```

continued

```
   SUBROUTINE ACTIONZ
   .
   .
   .
   END
7-8-9 (EOR Card)
Data Cards
6-7-8-9 (EOI Card)
```

NOTE: The file assignments on the Main-Program header card are the standard SPURT assignments.

SAMPLE INPUT

```
                SUBROUTINE TESTCOR
          C
          C     ****************************************************************
          C
          C     THIS PROGRAM IS PART OF THE NORTHWESTERN UNIVERSITY
          C
          C                     S  P  U  R  T
          C
          C        A SIMULATION PACKAGE FOR UNIVERSITY RESEARCH AND TEACHING
          C
          C     ****************************************************************
          C
          C
          C     ****************************************************************
          C
          C     THIS IS A TEST PROGRAM TO CHECK THE RESULTS OF THE
          C     STIX5(CORRELATION COEF.) ROUTINE OF THE SPURT  PACKAGE
          C     STIX6 IS ALSO TESTED
          C
          C     ****************************************************************
          C
000002          COMMON X(2000),Y(2000)
000002          DO 10 I=1,2000
000004          X(I)=RANF(-1)
000010    10    Y(I)=RANF(-1)
000016          PRINT 80
000021       80 FORMAT(5/,* TEST OF STIX5 AND STIX6*)
000021       20 FORMAT(3/,* COEFFICIENT OF CORRELATION BETWEEN *,A7,
               1* AND *,A7,* = *,F9.5)
000021          DUM=TIMELP(0)
000024          CALL STIX5(X,X,2000,R)
000027          N1=1HX
000030          N2=1HX
000032          PRINT 20,N1,N2,R
000044          DUM=TIMELP(0)
000047          CALL STIX5(X,Y,2000,R)
000052          N2=1HY
000054          PRINT 20,N1,N2,R
000065          DUM=TIMELP(1)
000070          CALL STIX6(X,2000,AMEDIAN,RANGE)
000073          PRINT 30,N1,AMEDIAN,RANGE
000105          DUM=TIMELP(1)
000110          CALL STIX6(Y,2000,AMEDIAN,RANGE)
000113          N1=1HY
000115          PRINT 30,N1,AMEDIAN,RANGE
000126          DUM=TIMELP(1)
000131          RETURN
          C
          C
          C
000131       30 FORMAT(  ///* FOR THE VARIABLE *,A7,* MEDIAN = *,F10.4,10X,
               1*RANGE = *,F10.4)
000131          END
```

SAMPLE OUTPUT

```
    *****     STIX4          *****          DATE  01/27/70      PAGE NO.  23


TEST OF STIX4

          MIN =      14.065
          MAX =      18.956
          SIGMA =     1.445
          MEAN =     16.631

     .. CP TIME SINCE LAST CALL ..    .024 SECONDS


TEST OF STIX5 AND STIX6

COEFFICIENT OF CORRELATION BETWEEN X        AND X        =   1.00000

COEFFICIENT OF CORRELATION BETWEEN X        AND Y        =   -.02223
     .. CP TIME SINCE LAST CALL ..    .129 SECONDS


FOR THE VARIABLE X       MEDIAN =      .4946          RANGE =      .9996
     .. CP TIME SINCE LAST CALL ..   1.303 SECONDS


FOR THE VARIABLE Y       MEDIAN =      .4809          RANGE =      .9979
     .. CP TIME SINCE LAST CALL ..   1.273 SECONDS
```

USER INSTRUCTIONS—SPURT4: Analog Simulators

The following two routines are designed to simulate an analog computer on a digital computer, thereby making it possible to obtain effects similar to those displayed by a hybrid. Both routines operate as integrators in an analog computer, integrating negatively with respect to time. Input (XIN) must be a floating-point number. Output (YOUT and TOUT) is two arrays, one giving values of time and the other giving the corresponding negative integral values. The output arrays are not printed out. Initial conditions Y may also be specified.

ANALOG

```
SUBROUTINE ANALOG(XIN,Y,YOUT,TOUT,T,DELTA,LIMIT,N)
TYPE REAL XIN,Y,YOUT,TOUT,T,DELTA
TYPE INTEGER LIMIT,N
```

Subroutine ANALOG is called in the following form: CALL ANALOG(XIN, Y,YOUT,TOUT,T,DELTA,LIMIT,N). The input value is XIN. YOUT and TOUT are the output arrays of integrated value and time, respectively. The user is cautioned to dimension YOUT and TOUT to the expected size in his program. Integration is carried out at discrete time intervals DELTA and for the number of intervals LIMIT. Any number of successive integrations N can be carried out. Thus, for N = 1, a first-order integration is performed; for N = 2, a second-order integration is performed; etc. Note that LIMIT and N are fixed-point numbers. T is the initial value of time, and TOUT(1) = T. Thus, TOUT(2) = T + DELTA and the last value TOUT(LIMIT) = T + DELTA *(LIMIT –1). Y is a vector of initial conditions of size N, where Y(1) is the initial value for the first-order integration, Y(2) for the second-order, etc. Thus, Y(1) gives the initial value of y; the output, Y(2), gives the initial value of $\int y\, dt$, etc. The array Y(N) must be dimensioned and provided in the user's program.

A second-order integration works exactly as do two successive integrators on an analog computer. Thus, for a positive input, the output array will also be positive.

SECND

```
SUBROUTINE SECND(XIN,YIC,DAMP,FREQ,T,YOUT,TOUT,DELTA,LIMIT)
TYPE REAL XIN,YIC,DAMP,FREQ,T,YOUT,TOUT,DELTA
TYPE INTEGER LIMIT
```

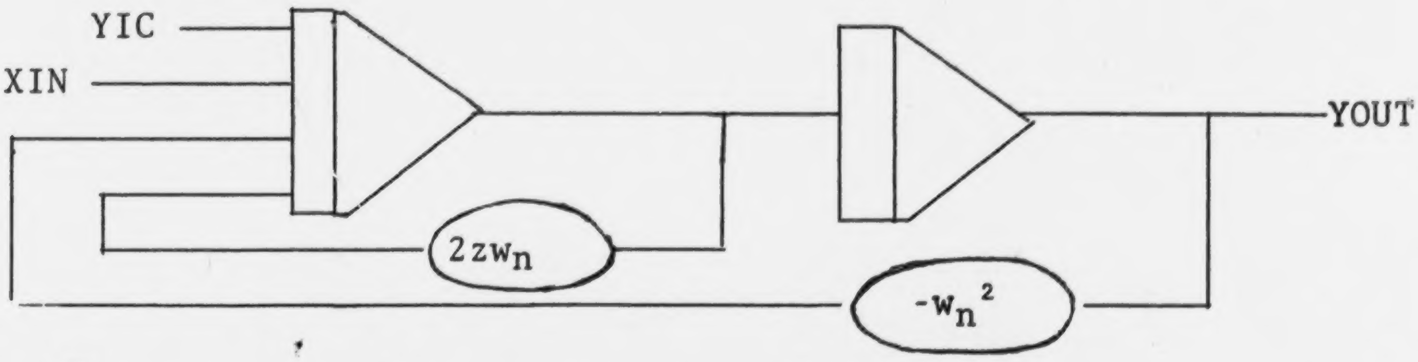
This routine provides the solution to the commonly used control equation

continued

SPURT4

$$F(x) = d^2y/dt^2 = 2zw_n(dy/dt) + w_n^2y,$$

where z is the damping coefficient, w_n is the natural frequency, and F(x) is a constant forcing function (such as a step input). SECND is equivalent to the analog connection



and is similar in operation to the subroutine ANALOG.

The form of the call is CALL SECND(XIN,YIC,DAMP,FREQ,T,YOUT,TOUT, DELTA,LIMIT), where T is the initial value of time, DELTA is the integration interval, LIMIT the number of values calculated, and YOUT and TOUT the output arrays. All are defiend as in subroutine ANALOG. Only LIMIT is fixed-point and YOUT and TOUT must be dimensioned in the user's program. YIC is the initial value of YOUT. Thus, YOUT(1) = YIC. XIN is the forcing function F(x). DAMP is the damping coefficient z. FREQ is the natural frequency w_n.

Job Deck Structure

```
Job Cards
LIBRARY(SPURT5,SPURT6)
LGO.
7-8-9 (EOR Card)
Main-Problem Card(s)
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
```

continued

```
    .
    .
    .
    SUBROUTINE ACTIONZ
    .
    .
    .
    END
7-8-9 (EOR Card)
Data Cards
6-7-8-9 (EOI Card)
```

NOTE: The file assignments shown on the Main-Program header card are the standard SPURT assignments.

SAMPLE INPUT/OUTPUT

The input for the Analog Simulators (SPURT4) will consist of floating-point numbers that are integrated negatively with respect to time. The output will be two arrays, one giving values of time and the other giving the corresponding negative integral values.

USER INSTRUCTIONS—SPURT5: List-Processing and Queue-Manipulation

ADFIFO,ADLIFO,REMOVE,PURGE,DISPL,IDISPL,NUMLST

```
SUBROUTINE ADFIFO(LIST,N,M,ENTRY)
SUBROUTINE ADLIFO(LIST,N,M,ENTRY)
SUBROUTINE REMOVE(LIST,N,M,ENTRY)
SUBROUTINE PURGE(LIST,N,M)
SUBROUTINE DISPL(LIST,N,M)
SUBROUTINE IDISPL(LIST,N,M)
INTEGER FUNCTION NUMLST(LIST)
INTEGER N,M
REAL (or INTEGER) LIST,ENTRY
```

The manipulations of queues may be handled by the four subroutines ADFIFO, ADLIFO, REMOVE, and PURGE. REMOVE will remove the top (or first) element from the list. ADLIFO inserts an element at the top of the list, which will be removed before any of the elements presently on the list. ADFIFO adds an entry at the bottom of the list, which will be removed only after all the elements presently on the list are gone. Lists may be manipulated by using both ADLIFO and ADFIFO on the same list array. A printout of the contents of a list may be achieved by use of either DISPL or IDISPL.

The general form of the calls to these routines is as follows.

```
CALL ADFIFO(LIST,N.M.ENTRY)
CALL ADLIFO(LIST,N,M,ENTRY)
CALL REMOVE(LIST,N,M,ENTRY)
CALL PURGE(LIST,N,M)
CALL DISPL(LIST,N,M)
CALL IDISPL(LIST,N,M)
NUM = NUMLST(LIST)
```

In the calling sequence, LIST is the user-defined array of size NXM and ENTRY is the user-defined vector, dimensioned (1,M), used as a buffer. As LIST(1,1) contains the list pointer, only N – 1 entries (each of size 1XM) are allowed in any list.

NOTE: Before loading the initial entry to *any* list, be sure to clear the word LIST(1,1) to zero.

The subroutines are designed to place the contents of a buffer (ENTRY) onto a user-defiend list. The buffer ENTRY should be filled by the user's program before ADFIFO or ADLIFO are called. In the

continued

case of REMOVE, the first entry is taken from the list and placed in the buffer ENTRY. The user may then use its contents.

The integer function NUMLST will return the number of entries in a given list. IQ = NUMLST(LIST3) will assign a value to IQ equal to the number of entries in LIST3.

RANKQS, IRNKQS, RANKQL, IRNKQL

```
SUBROUTINE RANKQS(LIST,N,M,J)
SUBROUTINE IRNKQS(LIST,N,M,J)
SUBROUTINE RANKQL(LIST,N,M,J)
SUBROUTINE IRNKQL(LIST,N,M,J)
INTEGER N,M,J
REAL or INTEGER LIST
```

Four routines are provided to rank queues formed by other SPURT list-processing routines. They may be used by coding of the following form.

```
CALL RANKQS(LIST,N,M,J)
CALL IRNKQS(LIST,N,M,J)
CALL RANKQL(LIST,N,M,J)
CALL IRNKQL(LIST,N,M,J)
```

where LIST is the user-defined array of size NXM and J $\leq$ M represents the "entry" position that is to be used to produce the desired ranking.

Subroutine RANKQS ranks floating-point numbers with the smallest first and IRNKQL ranks integer numbers with the largest first. Similarly, INRKQS ranks integer numbers in ascending order.

SEARCH, INSERT, DELETE

```
INTEGER FUNCTION SEARCH(LIST,N,M,J,ID)
SUBROUTINE INSERT(LIST,N,M,ENTRY,INDEX)
SUBROUTINE DELETE(LIST,N,M,ENTRY,INDEX)
INTEGER N,M,J,INDEX
REAL or INTEGER LIST,ENTRY,ID
```

Three routines are provided to facilitate the locating, deleting, and inserting of entries in the middle of a list. The use of the INTEGER SEARCH function in the form INDEX = SEARCH(LIST,N,M,J,ID), where SEARCH is of type INTEGER, will search LIST (of size NXM) in entry position J for the desired ID. If a match is found, SEARCH returns the position in the list of the first entry to contain the desired ID. If no match is found, SEARCH returns a zero value.

continued

Thus, SEARCH may be used within an arithmetic or logical IF statement to determine the presence or absence of a desired entry. The user may use either an integer or a real variable as the fifth argument in a call to SEARCH.

If the i^{th} entry has been located (say, by using SEARCH), then it may be removed by a call to DELETE in the form of CALL DELETE(LIST, N,M,ENTRY,INDEX), where INDEX is the number of the desired entry to be removed from LIST and placed in ENTRY.

To place a *new* i^{th} entry in the list, a call to INSERT in the form of CALL INSERT(LIST,N,M,ENTRY,INDEX) will create a new i^{th} entry (i is noted by INDEX).

Job Deck Structure

```
Job Cards
Run(s)
LIBRARY(SPURT4,SPURT6)
LGO.
7–8–9 (EOR Card)
Main-Problem Card(s)
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
   .
   .
   .
   SUBROUTINE ACTIONZ
   .
   .
   .
   END
7–8–9 (EOR Card)
Data Cards
6–7–8–9 (EOI Card)
```

SAMPLE INPUT

```
                   SUBROUTINE LIST1                                                      SPTS8  0
          C                                                                              SPTS8 10
          C        ***************************************************************SPTS8 20
          C                                                                              SPTS8 30
          C        THIS PROGRAM IS PART OF THE NORTHWESTERN UNIVERSITY                   SPTS8 40
          C                                                                              SPTS8 50
          C                          S   P   U   R   T                                   SPTS8 60
          C                                                                              SPTS8 70
          C          A SIMULATION PACKAGE FOR UNIVERSITY RESEARCH AND TEACHING           SPTS8 80
          C                                                                              SPIS8 90
          C        ***************************************************************SPTS8100
          C                                                                              SPTS8110
000002             COMMON A1,B1,C1,A,B,C                                                 SPTS8120
000002             INTEGER A,B,C,A1,B1,C1                                                SPTS8130
000002             DIMENSION A1(13,13), B1(12,17), C1(19,7)                              SPTS8140
000002             DIMENSION A(20,20), B(10,20), C(20,10), NMROWS(20), NMCOLS(20)        SPTS8150
000002             DIMENSION IENTRY(1,10)                                                SPTS8160
000002             DATA (NMROWS=2HR1,2HR2,2HR3,2HR4,2HR5,2HR6,2HR7,2HR8,2HR9,3HR10,3HSPTS8170
                  1R11,3HR12,3HR13,3HR14,3HR15,3HR16,3HR17,3HR18,3HR19,3HR20),(NMCOLSSPTS8180
                  2=2HC1,2HC2,2HC3,2HC4,2HC5,2HC6,2HC7,2HC8,2HC9,3HC10,3HC11,3HC12,3HSPTS8190
                  3C13,3HC14,3HC15,3HC16,3HC17,3HC18,3HC19,3HC20)                       SPTS8200
000002             C=0                                                                   SPTS8210
000003             M=0                                                                   SPTS8220
000004             DUM=TIMELP(0)                                                         SPTS8230
000006    10       I=0                                                                   SPTS8240
000007    20       I=I+10                                                                SPTS8250
000011             DO 30 J=1,10                                                          SPTS8260
000012             K=I+J                                                                 SPTS8270
000014    30       IENTRY(1,J)=K                                                         SPTS8280
000021             CALL INSOUT (IENTRY,1,10,NMROWS,NMCOLS)                               SPTS8290
000025             CALL ADFIFO (C,20,10,IENTRY)                                          SPTS8300
000030             CALL IDISPL (C,20,10)                                                 SPTS8310
000033             DUM=TIMELP(1)                                                         SPTS8320
000036             IF ( I .LT. 20) GO TO20                                               SPTS8350
000040             I=0                                                                   SPTS8360
000041    40       I=I+100                                                               SPTS8370
000043             DO 50 J=1,10                                                          SPTS8380
000044             K=I+J                                                                 SPTS8390
000046    50       IENTRY(1,J)=K                                                         SPTS8400
000053             CALL INSOUT (IENTRY,1,10,NMROWS,NMCOLS)                               SPTS8410
000057             CALL ADLIFO (C,20,10,IENTRY)                                          SPTS8420
000062             CALL IDISPL (C,20,10)                                                 SPTS8430
000065             DUM=TIMELP(1)                                                         SPTS8440
000070             IF ( I .LT. 200) GO TO 40                                             SPTS8450
000072             DO 60 I=1,4                                                           SPTS8460
000074             CALL REMOVE (C,20,10,IENTRY)                                          SPTS8470
000077             CALL INSOUT (IENTRY,1,10,NMROWS,NMCOLS)                               SPTS8480
000103             CALL IDISPL (C,20,10)                                                 SPTS8490
000106             DUM=TIMELP(1)                                                         SPTS8500
000111    60       CONTINUE                                                              SPTS8510
000113             RETURN                                                                SPTS8520
000113             END                                                                   SPTS8530
```

SAMPLE OUTPUT

```
*****     IDISPL        *****          DATE  01/27/70     PAGE NO.  33         NORTHWESTERN UNIVERSITY  S P U R T


   LIST DISPLAY ROUTINE INITIATED ...

                 OUTPUT IN FIXED POINT FORMAT


   ENTRY NUMBER     1
           11           12          13          14          15          16          17          18          19
              20


END OF OUTPUT BY IDISPL



   .. CP TIME SINCE LAST CALL ..    .042 SECONDS


*****     INSCUT        *****          DATE  01/27/70     PAGE NO.  34         NORTHWESTERN UNIVERSITY  S P U R T
              C1         C2         C3         C4         C5         C6         C7         C8         C9         C10
R1                21         22         23         24         25         26         27         28         29         30
```

continued

```
*****    IDISPL       *****        DATE  01/27/70     PAGE NO.  35        NORTHWESTERN UNIVERSITY  S P U R T


LIST DISPLAY ROUTINE INITIATED ...

              OUTPUT IN FIXED POINT FORMAT


ENTRY NUMBER     1
        11           12            13            14            15            16            17            18            19
            20

ENTRY NUMBER     2
        21           22            23            24            25            26            27            28            29
            30


END OF OUTPUT BY IDISPL



 .. CP TIME SINCE LAST CALL ..    .053 SECONDS


*****    INSCUT       *****        DATE  01/27/70     PAGE NO.  36        NORTHWESTERN UNIVERSITY  S P U R T

              C1          C2          C3          C4          C5          C6          C7          C8          C9          C10
R1                101         102         103         104         105         106         107         108         109         110
```

continued

SPURT5

```
*****     IDISPL        *****        DATE  01/27/70     PAGE NO.  37        NORTHWESTERN UNIVERSITY  S P U R T


  LIST DISPLAY ROUTINE INITIATED ...

                OUTPUT IN FIXED POINT FORMAT


  ENTRY NUMBER      1
        101           102           103           104           105           106           107           108           109
             110

  ENTRY NUMBER      2
        11            12            13            14            15            16            17            18            19
             20

  ENTRY NUMBER      3
        21            22            23            24            25            26            27            28            29
             30


END OF OUTPUT BY IDISPL


  .. CP TIME SINCE LAST CALL ..    .062 SECONDS
```

continued

000 0071

```
*****     IDISPL          *****         DATE  01/27/70     PAGE NO.  39        NORTHWESTERN UNIVERSITY  S P U R T


LIST DISPLAY ROUTINE INITIATFD ...

                OUTPUT IN FIXED POINT FORMAT


ENTRY NUMBER     1
       201          202         203         204         205         206         207         208         209
             210

ENTRY NUMBER     2
       101          102         103         104         105         106         107         108         109
             110

ENTRY NUMBER     3
       11           12          13          14          15          16          17          18          19
             20

ENTRY NUMBER     4
       21           22          23          24          25          26          27          28          29
             30


END OF OUTPUT BY IDISPL



  .. CP TIME SINCE LAST CALL ..    .070 SECONDS


 *****     INSOUT          *****         DATE  01/27/70     PAGE NO.  40        NORTHWESTERN UNIVERSITY  S P U R T

                 C1         C2         C3         C4         C5         C6         C7         C8         C9         C10
R1                 201        202        203        204        205        206        207        208        209        210
```

continued

000 0071

5 00

000 0071
SPURT5

```
*****    IDISPL        *****         DATE  01/27/70     PAGE NO.  41        NORTHWESTERN UNIVERSITY  S P U R T


LIST DISPLAY ROUTINE INITIATED ...

            OUTPUT IN FIXED POINT FORMAT


ENTRY NUMBER    1
      101         102        103        104        105        106        107        108        109
          110

ENTRY NUMBER    2
      11          12         13         14         15         16         17         18         19
          20

ENTRY NUMBER    3
      21          22         23         24         25         26         27         28         29
          30


END OF OUTPUT BY IDISPL



 .. CP TIME SINCE LAST CALL ..   .061 SECONDS


 *****    INSOUT       *****         DATE  01/27/70     PAGE NO.  42        NORTHWESTERN UNIVERSITY  S P U R T

               C1        C2        C3        C4        C5        C6        C7        C8        C9        C10
R1                101       102       103       104       105       106       107       108       109       110
```

continued

```
*****    IDISPL        *****        DATE  01/27/70     PAGE NO.  43         NORTHWESTERN UNIVERSITY  S P U R T


 [illegible] DISPLAY ROUTINE INITIATED ...

             OUTPUT IN FIXED POINT FORMAT


 ENTRY NUMBER      1
       11              12           13           14           15           16           17           18           19
           20

 ENTRY NUMBER      2
       21              22           23           24           25           26           27           28           29
           30


END OF OUTPUT BY IDISPL



  .. CP TIME SINCE LAST CALL ..    .053 SECONDS



*****    INSCOUT        *****        DATE  01/27/70     PAGE NO.  44         NORTHWESTERN UNIVERSITY  S P U R T

               C1          C2          C3          C4          C5          C6          C7          C8          C9          C10
R1                11          12          13          14          15          16          17          18          19          20
```

continued

```
*****    IDISPL       *****        DATE  01/27/70     PAGE NO.  45        NORTHWESTERN UNIVERSITY  S P U R T


  LIST DISPLAY ROUTINE INITIATED ...

           OUTPUT IN FIXED POINT FORMAT


  ENTRY NUMBER    1
        21          22         23         24         25         26         27         28         29
           30


 END OF OUTPUT BY IDISPL



  .. CP TIME SINCE LAST CALL ..   .045 SECONDS


*****    INSOUT       *****        DATE  01/27/70     PAGE NO.  46        NORTHWESTERN UNIVERSITY  S P U R T

          C1         C2         C3         C4         C5         C6         C7         C8         C9        C10
 R1           21         22         23         24         25         26         27         28         29         30
```

continued

```
*****     IDISPL         *****          DATE  01/27/70     PAGE NO.  47         NORTHWESTERN UNIVERSITY  S P U R T


    LIST DISPLAY ROUTINE INITIATED ...

                 OUTPUT IN FIXED POINT FORMAT


                 NO ENTRIES ON LIST



 END OF OUTPUT BY IDISPL



     .. CP TIME SINCE LAST CALL ..    .040 SECONDS
```

USER INSTRUCTIONS—SPURT6: Matrix and Graphical Output

OUT, IOUT

SUBROUTINE OUT(C,N,NAME)
SUBROUTINE IOUT(K,N,NAME)
TYPE REAL C
TYPE INTEGER N,K
Name is vector of Hollerith constants

Subroutine OUT prints out a square matrix with row and column headings. The form of the calling statement is CALL OUT(C,N,NAME), where C is the name of a two-dimensional square array, N is the number of rows (and columns) of the matrix C, and NAME is the name of a vector, dimensioned N in the user's program that contains the names to be printed over the columns (and along the rows). The heading of row j and column j will be the same, for j = 1, ···, N. The vector NAME must contain Hollerith constants of less than or equal to M (installation defined) characters defined in a DATA or assignment statement.

The matrix is printed out in sets of N rows by 10 columns. When the matrix printouts are placed beside one another, the entire matrix is represented.

The call to OUT assumes that *floating-point numbers* are to be printed out in E12.4 format. If the user wishes to print out *fixed-point numbers,* he should CALL IOUT(K,N,NAME), where K is an array of TYPE INTEGER and is to be printed out in I format.

NSOUT, INSOUT

SUBROUTINE NSOUT(C,NH,NV,NAMEH,NAMEV)
SUBROUTINE INSOUT(K,NH,NV,NAMEH,NAMEV)
TYPE REAL C
TYPE INTEGER NH,NV,K
NAMEH and NAMEV are vectors of Hollerith constants

Subroutine NSOUT is similar to subroutine OUT, except that NSOUT causes a nonsquare matrix, with appropriate row and column headings, to be printed out. The form of the CALL statement is CALL NSOUT(C, NH,NAMEH,NAMEV), where C is a two-dimensional array, NH is the number of rows of the array C, and NV is the number of columns of C. NAMEH is the name of a vector, dimensioned NH in the calling programs, that contains the names of the row headings. NAMEV is the name of a vector, dimensioned NV in the calling program, that contains the names of the column headings.

Calling the NSOUT subroutine causes the matrix C to be printed out

continued

in sets of N rows by 10 columns. The subroutine differs from OUT in that it allows column headings to differ from row headings.

The call to NSOUT assumes that *floating-point numbers* are to be printed out in E12.4 format. If the user wishes to print out *fixed-point numbers,* he should CALL INSOUT(K,NH,NAMEH,NAMEV), where K is an integer array to be printed out in I format.

GRAPH

SUBROUTINE GRAPH(NAMEX,LNTHX,NAMEY,LNTHY,X,Y,N)
TYPE REAL X,Y
TYPE INTEGER LNTHX,LNTHY,N
NAMEX and NAMEY are Hollerith constants

GRAPH produces two-dimensional graphs or plots on the CalComp plotter, requiring two one-dimensional arrays A and B of the same size, where $A = a_1, \cdots, a_n$, $B = b_1, \cdots, b_n$, and yielding a plot whose points are connected by straight line segments. When the user desires to print out a graph, he uses the following statement: CALL GRAPH(NAMEX,LNTHX,NAMEY,LNTHY,X,Y,N), where NAMEX is the label for the X axis. LNTHX is the *number of characters* in the title NAMEX. Similarly, NAMEY is the label for the Y axis and LNTHY is the number of characters in it. X and Y are the names of one-dimensional arrays (of the same size) to be plotted. N is the total number of points to be plotted, plus two. That is, if each array contains p entries, then N would be p + 2.

The following statement must appear immediately before all legal termination points in the user's program: CALL ENDPLT. This finalizes the output for the CalComp plotter. Note that no parameters are to be passed within the call to NEDPLT.

NOTE: The user may have titles of a length greater than M (installation defined) *only* if they are defined as Hollerith characters in the *calling sequence* to the GRAPH subroutine.

PAGE

SUBROUTINE PAGE(NAME)
NAME is a Hollerith constant

This routine is called to produce a page ejection in the printout of a program. It will inset up to M (installation defined) characters, found in NAME, into a heading that includes the date and page number. The most obvious use of NAME is to supply the name of the subroutine calling PAGE. For example, CALL PAGE(NAME). The routines in SPURT that produce page ejections (CLOCK, STIX2, etc.) all do so by calling

continued

PAGE. Thus, if a page number is to be meaningful to the user, he also must eject pages by a call to PAGE.

Job Deck Structure

```
Job Cards
LIBRARY(SPURT3,SPURT6)
LGO.
7–8–9 (EOR Card)
Main-Problem Card(s)
   PROGRAM(INPUT,OUTPUT,PUNCH,PLOT,TAPE5=INPUT, 1 TAPE6=OUTPUT,
   TAPE7=PUNCH,TAPE17=PLOT)
   .
   .
   .
   END
Subroutine Card(s)
   SUBROUTINE ACTION1
   .
   .
   .
   END
   .
   .
   .
   SUBROUTINE ACTIONZ
   .
   .
   .
   END
7–8–9 (EOR Card)
Data Cards
6–7–8–9 (EOI Card)
```

NOTE: The file assignments shown on the Main-Program header card are the standard SPURT assignments.

SAMPLE INPUT

```
                SUBROUTINE EXAMPL1                                                  SPTS2  0
         C                                                                          SPTS2 10
         C      ******************************************************************SPTS2 20
         C                                                                          SPTS2 30
         C      THIS PROGRAM IS PART OF THE NORTHWESTERN UNIVERSITY                 SPTS2 40
         C                                                                          SPTS2 50
         C                       S  P  U  R  T                                      SPTS2 60
         C                                                                          SPTS2 70
         C         A SIMULATION PACKAGE FOR UNIVERSITY RESEARCH AND TEACHING        SPTS2 80
         C                                                                          SPTS2 90
         C      ******************************************************************SPTS2100
         C                                                                          SPTS2110
000002          COMMON/QPLOTQ/UNUSED(5),IPLTSW                                      SPTS2120
000002          DIMENSION T(104)                                                    SPTS2130
000002          TYPE REAL NEGEXP                                                    SPTS2140
000002          DUM=TIMELP(0)                                                       SPTS2150
000005          CALL STIX1 (T,20.0,0.0,100,VALUE,22HDEPTH OF WATER IN TANK,22)      SPTS2160
000013          DO 10 I=1,300                                                       SPTS2170
000015          VALUE=20.0-NEGEXP(0.2)                                              SPTS2180
000020   10     CALL STIX2 (T,20.0,0.0,100,VALUE,22HDEPTH OF WATER IN TANK,22)      SPTS2190
000031          CALL STIX3 (T,20.0,0.0,100,VALUE,22HDEPTH OF WATER IN TANK,22)      SPTS2200
000040          DUM=TIMELP(1)                                                       SPTS2210
000043          RETURN                                                              SPTS2220
000043          END                                                                 SPTS2230
```

continued

000 0071
SPURT6

SAMPLE OUTPUT

```
****     STIX3          *****          DATE  01/27/70     PAGE NO.  20        NORTHWESTERN UNIVERSITY  S P U R T

                                  THIS DATA IS ACCOMPANIED BY GRAPH NUMBER        10

                DEPTH OF WATER IN TANK

                ENTRIES IN TABLE    MEAN ARGUMENT    STANDARD DEVIATION
                             300           14.665                 5.070

                UPPER        OBSERVED       PER CENT     CUMULATIVE     CUMULATIVE       MULTIPLE      DEVIATION     CUMULATIVE
                LIMIT       FREQUENCY       OF TOTAL     PERCENTAGE      REMAINDER        OF MEAN      FROM MEAN    PROBABILITY
UNDERFLOW
                0.000               6          2.000          2.000         98.000          0.000         -2.892           .002
                 .200               0          0.000          2.000         98.000           .014         -2.853           .002
                 .400               0          0.000          2.000         98.000           .027         -2.814           .002
                 .600               0          0.000          2.000         98.000           .041         -2.774           .003
                 .800               0          0.000          2.000         98.000           .055         -2.735           .003
                1.000               0          0.000          2.000         98.000           .068         -2.695           .004
                1.200               0          0.000          2.000         98.000           .082         -2.656           .004
                1.400               0          0.000          2.000         98.000           .095         -2.616           .004
                1.600               0          0.000          2.000         98.000           .109         -2.577           .005
                1.800               0          0.000          2.000         98.000           .123         -2.537           .006
                2.000               0          0.000          2.000         98.000           .136         -2.498           .006
                2.200               1           .333          2.333         97.667           .150         -2.459           .007
                            .             .              .              .
                            .             .              .              .
                            .             .              .              .
```

continued

000 0071

000 0071

```
    15.600     3   1.000   46.000   54.000   1.064    .184    .573
    15.800     4   1.333   47.333   52.667   1.077    .224    .589
    16.000     5   1.667   49.000   51.000   1.091    .263    .604
    16.200     5   1.667   50.667   49.333   1.105    .303    .619
    16.400     4   1.333   52.000   48.000   1.118    .342    .634
    16.600     7   2.333   54.333   45.667   1.132    .382    .649
    16.800     9   3.000   57.333   42.667   1.146    .421    .663
    17.000     3   1.000   58.333   41.667   1.159    .460    .677
    17.200     4   1.333   59.667   40.333   1.173    .500    .691
    17.400     7   2.333   62.000   38.000   1.186    .539    .705
    17.600     3   1.000   63.000   37.000   1.200    .579    .719
    17.800     7   2.333   65.333   34.667   1.214    .618    .732
    18.000     8   2.667   68.000   32.000   1.227    .658    .745
    18.200     6   2.000   70.000   30.000   1.241    .697    .757
    18.400    10   3.333   73.333   26.667   1.255    .737    .769
    18.600    11   3.667   77.000   23.000   1.268    .776    .781
    18.800     4   1.333   78.333   21.667   1.282    .815    .793
    19.000     8   2.667   81.000   19.000   1.296    .855    .804
    19.200     7   2.333   83.333   16.667   1.309    .894    .814
    19.400    13   4.333   87.667   12.333   1.323    .934    .825
    19.600    12   4.000   91.667    8.333   1.336    .973    .835
    19.800     8   2.667   94.333    5.667   1.350   1.013    .844
OVERFLOW  XXXXXXX    17   5.667  100.000     .000   1.364   1.052    .854
    20.200     0   0.000  100.000     .000   1.377   1.092    .862


        GRAPH NUMBER   10 HAS JUST BEEN PLOTTED ON CALCOMP

    THE PLOT IS OF ...

    DEPTH OF WATER IN TANK

         VS.

    FREQUENCY (PER CENT)


 END OF OUTPUT BY GRAPH
```

continued

COST ESTIMATE

For the SPURT TESGEN package listed on the Sample Input, the total running time was 91.59 seconds for the central-processor time and 46.597 seconds for the peripheral-processor time. Chargeable computer time was $14.16

Charge to user = computer time + postage and handling + network overhead
= $14.16 + $15.00 + network overhead
= $29.16 + network overhead

CONTENTS—SPURT

pages

1- 4 Abstract & Identification

User Instructions

5- 7 SPURT
9–13 SPURT1
21–25 SPURT2
35–38 SPURT3
41–43 SPURT4
47–49 SPURT5
61–63 SPURT6

I/O
SPURT

15–19 SPURT1
27–33 SPURT2
39–40 SPURT3
45 SPURT4
51–59 SPURT5
65–67 SPURT6
69 Cost—Contents

DESCRIPTIVE TITLE	FORTRAN Program to Assist in the Process of Political Reapportionment
CALLING NAME	T8 FSU BELOW
INSTALLATION NAME	Computer Center The Florida State University
AUTHOR(S) AND AFFILIATION(S)	William Below, Consultant Assembly Committee on Elections and Reapportionment California Legislature
LANGUAGE	CDC FORTRAN IV
COMPUTER	CDC 6400-65K
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Raymond Soller, Program Librarian/EIN Techncial Representative, Computing Center, Florida State University, Tallahassee, Fla. 32306 Tel.: (904) 599-4770

FUNCTIONAL ABSTRACT

BELOW is a program to assist with the development of political reapportionment plans using a set of procedures representative of the methods employed by legislatures to reapportion themselves. The actual algorithms utilized by the program are modifications of the computational techniques originally outlined by Stuart Nagel.[1]

The programmed procedures are those which follow. An initial district plan is determined by assigning geographic units to districts. This initial plan is fed into the processor along with demographic and political data for each unit. The program then tests each possible move of a unit from one district to another and then each trade of units between adjacent districts against a set of criteria. If the plan is approved according to the criterion; then the move or trade is made permanent. Where no more useful moves or trades can be made, the results are printed out and the program is terminated.

continued

The criterion used by the program is expressed as a linear combination of three quantities, SVEA, SVCA, SVPA;

$$\text{CRIT} = \text{WE} \cdot \text{SVEA} + \text{WC} \cdot \text{SVCA} + \text{SVPA}$$

where SVEA, SVCA, and SVPA are measures of population equality, compactness, and conformance to political goals, and WE and WC are weighting coefficients for the purpose of establishing the relative importance of the three quantities. CRIT, the criterion, is recomputed for each tentative move or trade. If its value is less than its previous value, the move or trade is consummated.

Population Equality

The measure of population equality is given by

$$\text{SVEA} = \sum_{J=1}^{\text{NDA}} \left[\frac{[\text{PD (J)} - \text{AVGPOP (J)}]}{\text{AVGPOP (J)}} \cdot 100 \right]^2$$

where NDA is the number of districts in the area, PD (J) is the population of district J and AVGPOP (J) is the ideal population. SVEA may be described as the sum of the squared fractional deviations from the ideal population for all districts in the area.

Multiplication of the fractional deviation by 100 is not significant to the operation of the program; it simply indicates that the deviation is output as a percentage. The use of the fractional rather than the absolute deviation is significant because it makes the effect of WE, (the relative importance of population equality), independent of the ideal population. Similar considerations apply to the decision to use total rather than average deviation for the area. If average deviation were used, then the proper setting of WE would depend on the number of districts in the area.

The opportunity for the user to set the ideal population for each district independently may at first seem arbitrary. In practice, this allows the program to be used on areas which contain portions of districts as well as whole districts.

This method of measuring population equality does not differ markedly from methods used by others in the field. One obvious alternative, namely, use of the sum of the absolute deviations, might be quite acceptable as a measure of equality to a student government, but it suffers from some computational disadvantages.

continued

As a population unit moves from one district to another, the absolute deviation changes only if one district is above, while the other is below the ideal population. The formula given here for SVEA however, has the convenient property that when two districts are both above or below the ideal population, a unit moved from one district to the other causes a change in SVEA, proportional to the difference in population between the two districts.

Compactness

The method originally tried for measuring compactness determines the "population moment" around the population center of the district. This method is attractive because it tends to minimize travel distance within the district and keep concentrations of population unbroken by district lines.

If there is a large number of units per district, the population moment method, unfortunately, can yield districts which are numerically compact, yet have very irregular boundaries. Since a legislature, as it reapportions itself, prefers to avoid even the appearance of gerrymandering, the author found the population moment method unsatisfactory for his purposes.

The alternative method, now in use, involves estimation of the perimeter of each district. The estimation is inherently inexact, but tends to produce district lines which are politically acceptable.

The program counts the perimeter elements in the perimeter of each district. A perimeter element is that portion of a unit perimeter which is shared with one other unit. Of course, these elements vary in length, but the program assumes each to have unit length.

One of the attractive features of the perimeter method is that it dispenses with the use of the X and Y coordinates required for the population moment method. What may be unattractive to some users is that the perimeter method does not tend to improve or even affect the overall shape of a district unless the initial district plan deviates widely from the desired characteristics.

Thus, if the program is given a district which is rectangular, and which is much longer than it is wide, the lines will be kept straight, but there will be little or no tendency for that district to change toward a square or circular shape. This property is often quite convenient in the legislative environment.

continued

In principle, it would be possible to measure the length of each perimeter element, include the data in the information fed to the processor, and work with a true perimeter measurement. Making those measurements, however, would be even more tedious than preparing the X and Y data, and may not necessarily improve program performance.

Political Considerations

The political portion of the criterion, CRIT, is given by

$$\mathrm{SVPA} = \sum_{J=1}^{\mathrm{NDA}} \mathrm{WP}\ (J) \left[\frac{\mathrm{PAD}\ (J) \cdot 100}{\mathrm{PBD}\ (J)} - \mathrm{DESPR}\ (J) \right]^2$$

where WP (J) is a political weighting coefficient that may be set independently for each district, PAD (J) and PBD (J) are political quantities for district and DESPR (J) is the desired ratio of PAD (J) to PBD (J) expressed as a percentage.

PAD (J) and PDB (J) may stand for different things in different districts according to a number called MODE (J). If MODE (J) were set to 1, for example, and DESPR (J) to 55, then the program would set PAD (J) to equal registered Democrats and PBD (J) to equal total registered voters, and the goal for Democratic registration in that district would be 55%.

In the same manner, setting MODE (J) to 2 or 3 will establish a goal for the percentage of Negroes or persons with a Spanish surname in the population. The information for each unit necessary to establish any of the three proportions is carried in the memory of the processor. Consequently, there are no restrictions on moving a unit between districts with different modes.

Constraints on Moving and Trading

By the inclusion of the one card per unit in the data deck, the user may forbid the program to move or trade any number of units out of their original districts. The most common use of this feature is to ensure that incumbents are left in their own districts. Another use is to protect the integrity of municipal areas.

REFERENCES

1. Nagel, S., "Simplified Bipartisan Computer Redistricting," *Stanford Law Review, 17*, No. 5, May 1965.

2. Below, W., "The Computer as an Aid to Legislative Reapportionment," (Report to California Legislature, Assembly Committee on Elections and Reapportionment).

Unit Card

One Unit Card is needed for each Census tract, enumeration district, precinct, or other geographical unit.

Columns	*Identification*	*Contents*
1		1: Unit Card label
2– 7	UNIT CODE	alphanumeric unit code used to identify a unique geographic unit.
8–11	PU	population count within the unit
12–15	P1	population characteristic one
16–19	P2	population characteristic two
20–23	P3	population characteristic three
24–27	P4	population characteristic four
		In a single pass of the program the selected data that is tabulated into district totals (P1/P2, P3/POPULATION, P4/POPULATION) is determined by the value of MODE
38–79 partioned into 7 fields of 6 columns	TOUCHLIST	listing of all unit codes which are adjacent to the unit identified in Cols. 2–7.
80	NL	number of neighbors entered into TOUCHLIST field

The TOUCHLIST is used for establishing and maintaining contiguity, so all rules and assumptions concerning contiguity should be considered when compiling the TOUCHLIST. The TOUCHLIST is also used to measure compactness. The computations are more precise if reciprocity is observed. Reciprocity implies that if Unit A is a member of the TOUCHLIST for Unit B, then Unit B is also a member of the TOUCHLIST for Unit A.

TOUCHLIST Continuation Card

If there are more than seven neighbors in a unit's TOUCHLIST the remaining neighbors may be inserted on a Continuation Card. A unit may have as many Continuation Cards as needed.

continued

Columns	*Identification*	*Contents*
1		2: Unit Continuation Card label
38–79	TOUCHLIST	listing of unit code neighbors continued from previous card
80	NL	number of neighbors entered into the TOUCHLIST field

The total number of entries for the complete TOUCHLIST must not exceed 8500.

"Eight" Card

This card signals the end of the initial district data.

Columns	*Identification*	*Contents*
1		8: "Eight" Card label

Status Card

A Status Card is necessary if the possible moves or trades of a unit out of its district are to be restricted. One card is required required for each such specified unit, and must follow an "Eight" Card.

Columns	*Identification*	*Contents*
1		4: Status Card label
2– 7	UNIT CODE	alphanumeric unit code used to identify a unique geographic unit.
11		1: flag which restricts unit movements

"Nine" Card

This card initiates the district processing, and must follow the status cards if present or an "Eight" Card.

Columns	*Identification*	*Contents*
1		9: "Nine" Card Label

Parameter Update Card

An update pass is possible by changing the label field (Col. 1) to a 5 and by adjusting the proper parameters. All entries must be present even if they repeat the items on the Parameter Card. If it is desired to end the restrictions on a unit, then include a Status Card with a zero in Col. 11.

continued

Any number of reruns can be made, but each successive run will continually distort its relation to the initial district plan.

Sense Switches

Sense switches can be set from the console by the machine operator or by a SWITCH CARD in the set of system control cards accompanying the job.

SWITCH, n.

where n is a switch from 1 to 6, which should be set on.
sense switch 2 on - suppress most printer output.
sense switch 3 on - (by operator intervention only) Terminates program in orderly manner.
sense switch 5 on - suppress trading mode of operation which normally begins when no more moves can be made

Output

The printed output of the program consists of a one page report for each district and an area summary for the initial plan, for the final plan, and for each successive rerun.

Each district report includes a complete list of the units in the district, a partial list showing only units on the edge of the district (for map drawing), and a numerical description of the district. The quantities "A" and "B" correspond to two of the quantities P1, P2, P3, or P4 according to the mode setting.

One line is printed summarizing the results of each major program loop. These lines appear between the initial and final district report pages. The numbers under NTM and NTT indicate the number of proposed moves and trades in each loop which satisfy the contiguity requirements and hence the number of times the program sections which tentatively recompute the criterion function are entered. These loop summaries are also printed on-line unless sense switch 2 is down.

Error Message

An error condition caused by improper data card punching results in an error message printed both off-line and on-line followed by exit. The message is head by "DATA ERROR" and consists of the following items.

continued

Item	*Contents*
IRD (1)	Col. 1 of last card read
IRD (2)	Cols. 2–7 of last card read
I	Internal unit counter
J	Internal district counter
K	Internal TOUCHLIST counter
M	See program listing
LPTYP	See program listing
JRET	See program listing

"Not in Area" Message

Each TOUCHLIST entry which does not have a corresponding Unit Card is listed under the heading "ON TOUCHLIST, NOT IN AREA." Two hundred such entries can be accomodated by the program. The presence of unmatched TOUCHLIST entries does not constitute an error condition and is common when a subset of the Unit Cards for an area is used for a program run.

"Not Contiguous" Message

A check for contiguity is made on the initial district plan. If any initial district is not contiguous a message of the form "DISTRICT J NOT CONTIGUOUS" is printed. The program does not stop since this condition is not necessarily disastrous to the program run. The operator may, however, be instructed to terminate execution when this message appears on-line.

SAMPLE INPUT

```
0        100  25 400   10   -0    -0   -0L.A. CONGESSIONAL REAPPORTIONMENT - BELOW *
3          1 1004376    1    1    -0   -0                                          *
1      1 7021766 8601011    1    -0   -0     6     9     2     3   635  1296      6
1      2 500 803 4981012    1    -0   -0     1     3     4                        3
1      3 4511114 3701013    1    -0   -0     1   635   636     5     4     2      6
1      4 284 961 3921014    1    -0   -0     9     2     3     5                  4
1      5 6591677 7681021    1    -0   -0    75    65     8     9     4   636   6397
2         -0  -0  -0  -0   -0    -0   -0   658   661     3                        3
1      6 8021710 8791031    1    -0   -0     7     9     1  1297                  4
1      7 4011098 5761032    1    -0   -0     8     9     6    10  1297            5
1      8 3811376 4301033    1    -0   -0    10    65     5     9     7            5
1      9 5721554 6741034    1    -0   -0     5     4     1     6     7     8      6
1     351310469622141112    1    -0   -0    38    40    44    36    34    23  12947
1     36 4991645 9941113    1    -0   -0    35    43    42    37                  4
1     38 201 706 2921131    1    -0   -0    39    40    35  1294                  4
1     39 947299412831132    1    -0   -0    38    40    41   140   141   142  12947
1     40 859237111461133    1    -0   -0    39    41    44    35    38            5
1     411518253111231134    1    -0   -0    39   139   125    44    40            5
1     42 5421657 7281151    1    -0   -0    36    43    45    48                  4
1     43 705190710101152    1    -0   -0    36    44    45    42                  4
1     44 1721579 8701153    1    -0   -0    40    41   125    45    43    35      6
1     45 6631626 8681154    1    -0   -0    44   124   123   122   121   120    487
2         -0  -0  -0  -0   -0    -0   -0    42    43                              2
                  .                          .                     .
                  .                          .                     .
                  .                          .                     .
1   1149 168  -0  -05990   16    -0   -0  1147  1150                              2
1   1150  39  -0  -05991   16    -0   -0  1149  1147                              2
3          17 1004376   1    1    -0   -0                                          *
1   10721411324117265700   17    -0   -0  1063  1062  1073  1079  1080  1081  10647
1   1080 6951382 8435708   17    -0   -0  1072  1079  1084  1081                  4
1   10811144265416585709   17    -0   -0  1072  1080  1083  1082  1064            5
1   1082 847175611835710   17    -0   -0  1081  1083  1109  1110  1069            5
1   10831199249714925711   17    -0   -0  1081  1084  1108  1109  1082            5
1   1109 700   0   05737   17    -0   -0  1082  1083  1108  1113  1110            5
1   1110 683   0   05738   17    -0   -0  1082  1109  1112  1111  1069            5
1   1111  16   0   05739   17    -0   -0  1110  1112  1146  1117  1071  1069      6
8         -0  -0  -0  -0   -0    -0   -0                                          *
9         -0  -0  -0  -0   -0    -0   -0                                          *
```

SAMPLE OUTPUT

```
DISTRICT   1                                          L.A. CONGESSIONAL REAPPORTIONMENT - BELOW
INITIAL


UNITS

      1          2          3          4          5          6          7          8          9         35
     36         38         39         40         41         42         43         44         45         75
     92         93         94         95         96         97         99        100        101        117
    119        125        126        133        134        139        140        141        142        143
    144        145        146        148        149        164        165        166        169        170
    171        172        173        174        175        176        178        179        662        665
    666        667        668        669        670        671        672        673        674        675
    676       1268       1269       1270       1271       1272       1273       1274       1275       1276
   1277       1278       1279       1280       1281       1282       1283       1284       1285       1286
   1287       1288       1289       1290       1291       1292       1293       1294       1297

UNITS ON EDGE OF DISTRICT

      1          3          5          7          8         35         36         42         45         75
     92         93         94         95         96         97         99        100        101        117
    119        125        126        133        134        143        144        146        148        149
    166        169        170        176        178        179        662        665        666        668
    669        671        673        674        675        676       1288       1290       1294       1297


       POPULATION            VARIATION        PCT VARIATION       SQRD VARIATION            WEIGHTED

       50625.0000            6865.0000              15.6878             246.1084            246.1084

            A                    B               PROPORTION       SQRD VARIATION            WEIGHTED

      123162.0000           68169.0000             180.6716            6507.9016             65.0790

       NO OF UNITS                 WP                DESPR          COMPACTNESS            WEIGHTED

          99.0000                .0100             100.0000              98.0000            392.0000


       TOTAL CONTRIBUTION TO CRITERION  =        703.1874
```

continued

```
DISTRICT   2                                    L.A. CONGESSIONAL REAPPORTIONMENT - BELOW
INITIAL


UNITS

   10        11        12        13        14        15        16        17        18        19
   20        21        22        23        24        25        26        27        28        29
   30        31        32        33        34        37        46        47        48        49
   50        51        52        53        54        55        56        57        58        59
   60        61        62        63        64        65        66        67        68        69
   70        71        72        73        74        76        77        78        79        80
   81        82        83        84        85        86        87        88        89        90
   91        98       102       103       104       105       106       107       108       109
  110       111       112       113       114       115       116       118       120       121
  122       123       124       127       128       129       130       131       132       135
  136       677       678       679

UNITS ON EDGE OF DISTRICT

   10        11        18        19        20        21        22        23        34        37
   49        65        74        76        78        83        85        86        87        89
   90        91        98       113       115       116       118       120       121       122
  123       124       132       135       136


      POPULATION          VARIATION        PCT VARIATION       SQRD VARIATION          WEIGHTED

      51716.0000          7956.0000              18.1810             330.5483          330.5483


           A                  B               PROPORTION       SQRD VARIATION          WEIGHTED

      90056.0000         68748.0000             130.9944             960.6501            9.6065


      NO OF UNITS             WP                 DESPR           COMPACTNESS           WEIGHTED

        104.0000            .0100             100.0000              62.0000          248.0000


      TOTAL CONTRIBUTION TO CRITERION  =         588.1548
```

⋮ ⋮ ⋮

continued

```
DISTRICT  17                                    L.A. CONGESSIONAL REAPPORTIONMENT - BELOW
INITIAL


UNITS

  1072      1080      1081      1082      1083      1109      1110      1111

UNITS ON EDGE OF DISTRICT

  1072      1080      1081      1082      1083      1109      1110      1111

         POPULATION          VARIATION          PCT VARIATION       SQRD VARIATION          WEIGHTED

          6595.0000        -37065.0000             -84.7005            7174.1984            7174.1984

             A                  B                PROPORTION         SQRD VARIATION          WEIGHTED

         11530.0000          6902.0000             167.0530            4496.1086              44.9611

         NO OF UNITS             WP                 DESPR            COMPACTNESS            WEIGHTED

             8.0000            .0100              100.0000              20.0000              80.0000


         TOTAL CONTRIBUTION TO CRITERION  =         7299.1595
   1     2     3     4     5     6     7     8     9    35    36    38    39    40    41    42    43    44    45    75
  92    93    94    95    96    97    99   100   101   117   119   125   126   133   134   139   140   141   142   143
 144   145   146   148   149   164   165   166   169   170   171   172   173   174   175   176   178   179   662   665
 666   667   668   669   670   671   672   673   674   675   676  1268  1269  1270  1271  1272  1273  1274  1275  1276
1277  1278  1279  1280  1281  1282  1283  1284  1285  1286  1287  1288  1289  1290  1291  1292  1293  1294  1297    10
  .                                     .                                                                 .
  .                                     .                                                                 .

 993   994   995   996   997   998   999  1000  1001  1002  1024  1025  1026  1027  1028  1029  1030  1031  1032  1033
1034  1035  1036  1037  1050  1051  1052  1053  1054  1055  1056  1057  1058  1059  1060  1061  1063  1062  1073  1074
1075  1076  1077  1078  1079  1084  1085  1086  1087  1088  1089  1090  1091  1092  1093  1094  1095  1096  1097  1098
1099  1100  1101  1102  1103  1104  1105  1106  1107  1108  1112  1113  1114  1115  1116  1117  1118  1119  1120  1121
1122  1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135  1136  1137  1138  1139  1140  1141
1142  1143  1144  1145  1146  1147  1148  1149  1150  1072  1080  1081  1082  1083  1109  1110  1111


CHARACTERISTICS OF THE AREA                          L.A. CONGESSIONAL REAPPORTIONMENT - BELOW

INITIAL


                TOT VAR          WEIGHT          VAR X WEIGHT


EQUALTIY      20392.66467         1.00000         20392.66467


POLITICAL                                          1073.13172


COMPACTNESS    1219.00000         4.00000          4876.00000



CRITERION  =     26341.79639
```

continued

```
LOOP      MOVES     TRADES   COMPACTNESS      EQUALITY     POLITICAL     CR2TERION        NTM       NTT

   0          0          0   1219.00000   20392.66467    1073.13172   26341.79639
LOOP      MOVES     TRADES   COMPACTNESS      EQUALITY     POLITICAL     CRITERION        NTM       NTT

   1        158          0   1289.00000    9643.07913     948.05748   15747.13661        606         0
   2        120          0   1329.00000    5153.01916     930.84078   11399.85994        560         0
   3        105          0   1379.00000    2178.37392     917.28298    8611.65690        556         0
   4         93          0   1385.00000     827.30062     907.10967    7274.41029        565         0
   5         68          0   1339.00000     253.83383     905.14882    6514.98265        555         0
   6         45          0   1283.00000     121.17347     920.09572    6173.26920        555         0
   7         27          0   1245.00000      95.37615     919.89828    5995.27443        553         0
   8         26          0   1217.00000      60.09501     916.89478    5844.98979        545         0
   9         13          0   1211.00000      59.23051     909.21067    5812.44119        537         0
  10         17          0   1187.00000      66.48433     919.32627    5733.81060        534         0
  11         13          0   1169.00000      69.77466     920.85634    5666.63100        533         0
  12         15          0   1155.00000      66.44527     929.61987    5616.06513        533         0
  13         16          0   1143.00000      51.90356     923.19476    5547.09832        522         0
  14          6          0   1139.00000      45.85531     922.50439    5524.35971        525         0
  15         13          0   1123.00000      49.64380     922.35721    5464.00101        514         0
  16          3          0   1123.00000      43.31775     921.33409    5456.65184        516         0
  17          4          0   1121.00000      42.50554     919.94415    5446.44969        514         0
  18          6          0   1113.00000      42.82931     919.54397    5414.37328        514         0
  19          0          0   1113.00000      42.82931     919.54397    5414.37328        513         0
  20          5         16   1091.00000      52.48909     915.98888    5332.47798        519      2498
  21         18          9   1059.00000      42.18904     917.03800    5195.22704        514      2367
  22         12         13   1023.00000      35.10415     942.11087    5069.21502        498      2217
  23         13          9   1011.00000      21.63398     927.43102    4993.06500        485      2150
  24          9          6    999.00000      24.35529     920.70191    4941.05720        480      2112
  25         12          3    983.00000      32.17103     911.84154    4876.01257        479      2073
```

continued

```
DISTRICT   1                                      L.A. CONGESSIONAL REAPPORTIONMENT - BELOW
FINAL


UNITS


      5         6         7         8         9        75        92        93        94        95
     96        97        99       100       101       171       172       173       174       175
    176       178       179       662       665       666       667       668       669       670
    671       672       673       674       675       676      1268      1269      1270      1271
   1272      1273      1274      1275      1276      1277      1278      1279      1280      1281
   1282      1283      1284      1285      1286      1287      1288      1289      1290      1291
   1292      1293      1297        10        11        12        16        18        19        20
     21        22        24        25        65        73        74        76        77        78
     79        85        86        87        88        89        90        98       677       678
    177       658       659       660       661       663       664

UNITS ON EDGE OF DISTRICT

      5         6         9        92        93       178       179       673       674       675
   1288      1290      1291      1292      1297        12        16        22        24        65
     73        77        79        85        90       678       177       658       659       664


       POPULATION            VARIATION         PCT VARIATION        SQRD VARIATION          WEIGHTED

       44359.0000             599.0000                1.3698                1.8737            1.8737

            A                     B               PROPORTION        SQRD VARIATION          WEIGHTED

       98999.0000           59135.0000              167.4102             4544.1301           45.4413

       NO OF UNITS               WP                   DESPR           COMPACTNESS           WEIGHTED

          97.0000                .0100              100.0000               53.0000          212.0000


       TOTAL CONTRIBUTION TO CRITERION  =        259.3150
```

.   .   .
.   .   .
.   .   .

continued

```
DISTRICT  17                                  L.A. CONGESSIONAL REAPPORTIONMENT - BELOW
FINAL


UNITS


   1018      1038      1039      1040      1041      1044      1045      1046      1047      1048
   1049      1064      1065      1066      1067      1068      1069      1070      1071      1037
   1050      1051      1052      1053      1054      1058      1059      1060      1061      1063
   1062      1073      1074      1075      1077      1078      1079      1084      1085      1086
   1087      1088      1089      1090      1091      1092      1093      1094      1096      1104
   1105      1106      1107      1108      1112      1113      1114      1115      1116      1117
   1072      1080      1081      1082      1083      1109      1110      1111

UNITS ON EDGE OF DISTRICT

   1018      1038      1039      1041      1044      1064      1037      1053      1054      1058
   1074      1075      1089      1094      1096      1104      1105      1106      1114      1115
   1117      1111


      POPULATION          VARIATION         PCT VARIATION      SQRD VARIATION         WEIGHTED

      43462.0000         -298.0000             -.6810               .4637               .4637

           A                  B               PROPORTION        SQRD VARIATION         WEIGHTED

      69513.0000         42115.0000           165.0552           4232.1798            42.3218

      NO OF UNITS            WP                 DESPR            COMPACTNESS           WEIGHTED

         68.0000            .0100             100.0000             42.0000            168.0000


      TOTAL CONTRIBUTION TO CRITERION  =        210.7855
    1     2     3     4     5     6     7     8     9    35    36    38    39    40    41    42    43    44    45    75
   92    93    94    95    96    97    99   100   101   117   119   125   126   133   134   139   140   141   142   143
  144   145   146   148   149   164   165   166   169   170   171   172   173   174   175   176   178   179   662   665
  666   667   668   669   670   671   672   673   674   675   676  1268  1269  1270  1271  1272  1273  1274  1275  1276
 1277  1278  1279  1280  1281  1282  1283  1284  1285  1286  1287  1288  1289  1290  1291  1292  1293  1294  1297    10
   11    12    13    14    15    16    17    18    19    20    21    22    23    24    25    26    27    28    29    30
   31    32    33    34    37    46    47    48    49    50    51    52    53    54    55    56    57    58    59    60
         .                                         .                                                       .
         .                                         .                                                       .
         .                                         .                                                       .
  983   994   995   996   997   998   999  1000  1001  1002  1024  1025  1026  1027  1028  1029  1030  1031  1032  1033
 1034  1035  1036  1037  1050  1051  1052  1053  1054  1055  1056  1057  1058  1059  1060  1061  1063  1062  1073  1074
 1075  1076  1077  1078  1079  1084  1085  1086  1087  1088  1089  1090  1091  1092  1093  1094  1095  1096  1097  1098
 1099  1100  1101  1102  1103  1104  1105  1106  1107  1108  1112  1113  1114  1115  1116  1117  1118  1119  1120  1121
 1122  1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135  1136  1137  1138  1139  1140  1141
 1142  1143  1144  1145  1146  1147  1148  1149  1150  1072  1080  1081  1082  1083  1109  1110  1111
```

continued

```
   CHARACTERISTICS OF THE AREA                            L.A. CONGESSIONAL REAPPORTIONMENT - BELOW

   FINAL

                     TOT VAR            WEIGHT        VAR X WEIGHT

   EQUALTIY         32.17103           1.00000         32.17103

   POLITICAL                                          911.84154

   COMPACTNESS     983.00000           4.00000       3932.00000

   CRITERION  =      4876.01257
*        -0  -0  -0  -0  -0   -0   -0                                       *
DATA ERROR

*           1297   17    1   68    1    2
```

COST ESTIMATE

Computer charges are calculated at the rate of $150./computational unit (hour). The sample run consumed 7 minutes of computer time, which should be typical for large jobs of similar complexity.

Therefore, for the sample run shown above,

Approximate charge to user = computer time + postage, handling, and consulting + network overhead
= $17.50 + $7.50 + network overhead
= $25.00 + network overhead

CONTENTS—T8 FSU BELOW

pages

1– 4 Identification & Abstract
5–10 User Instructions
11–18 I/O
19 Cost—Contents

DESCRIPTIVE TITLE	Equipercentile Equating Program
CALLING NAME	SCORMACH
INSTALLATION NAME	Wharton Computational Services University of Pennsylvania
AUTHOR(S) AND AFFILIATION(S)	Daniel Ashler, Director Wharton Computational Services University of Pennsylvania
LANGUAGE	FORTRAN IV
COMPUTER	IBM System 360/75
PROGRAM AVAILABILITY	Proprietary; usage permitted but program deck or listing not available
CONTACT	Daniel Ashler, Wharton Computational Services, University of Pennsylvania, Philadelphia, Pa. 19104 Tel: (215) 594-6422

FUNCTIONAL ABSTRACT

SCORMACH uses the equipercentile equating method to obtain "comparable" scores (having identical means, standard deviations, and distributions) for several forms of a given test or for different tests. A given score on one test is considered comparable to a given score on another if, in the two (not necessarily distinct) groups of examinees, the same proportions attain less than the respective given scores—i.e., if a score on one test has the same percentile rank as its comparable score on the other test.

The principal output is a table for each pair of tests by which, given a regressed equipercentile score on either test, one can determine the corresponding regressed equipercentile score on the other. If desired, the same information may also be had in the form of a graph produced on a CalComp plotter. Other tables and graphs allow comparisons of raw scores with raw scores, raw scores with regressed scores, etc.

The scores that are of interest here are the examinees' "true scores" on each respective test. Although true scores cannot be observed directly, the characteristics of the distribution

continued

of true scores for a given test can be estimated. This estimation, resulting in "regressed" scores, is made by modifying the value of each observed score in accordance with the reliability coefficient of the test.[1] One technique for equating scores is to plot points that correspond to pairs of comparable regressed scores on rectangular coordinates. A smooth curve is drawn through the points; pairs of equated scores are read from this curve. In addition to equating scores, the program predicts scores on one test from raw scores on the other. The predicted score is the regressed score that has the same percentile rank as the predictor raw score on the other test.

The program accepts a set of raw scores and a coefficient of reliability for each of two or more tests. (There is no limit to the number of tests to be equated in a single run.) The scores of the first test (*anchor test*) entered as input are equated with the scores of each subsequent test, one test at a time, by the equipercentile method described above.

The comparability of the scores of different forms (or tests) is specific to the type of groups used in obtaining them. A group that is both representative (of the population of examinees for whom the tests are intended) and large (at least 500 examinees) should be used. Preferably, the same examinees should take all tests to be equated (Ref. 1, p. 758). Although the value of the results is enhanced when all tests are taken by the same examinees and by an equal number of examinees, the program does not require that these conditions be met.

REFERENCES

1. Flanagan, J.C., "Units, Scores, and Norms," in *Educational Measurement*, E.F. Lindquist, Ed. (American Council on Education, Washington D.C., 1951), pp. 752–760.

USER INSTRUCTIONS

Preparation of Input

System Cards

Wharton Computational Services will prepare the necessary system cards to conform with the current version of the operating system. A cover letter should accompany the data stating whether CalComp output is desired.

Data-Input Cards

A. Provide a score packet for each test to be analyzed, consisting of a Title Card followed by Score Cards for that test. Data for the "anchor" test should come first. Scores of each subsequent test are equated with those of the "anchor" test. There is no limit to the number of tests for which score packets are submitted.

(1) Title Card

Columns	*Contents*
1– 8	Alphabetic title of test. It may *not* include any numeric digits, periods, plus signs or minus signs
9–80	Ignored by the program

(2) Score Cards

Punch as many test scores as desired on each card. Scores must be separated by at least one blank. When Col. 80 of a card is reached, continue in Col. 1 of the next card. (If a score ends in Col. 80, begin the next score in Col. 2 of the next card to have a blank between scores.) The scores need not be in ascending order *but computer time will be saved* if they are. If a score is repeated N times, the score need not be punched N times. Instead, punch the value of N followed by an asterisk followed by the score. For example, if 17 people have the score 112.5, punch 17*112.5. Follow the last score with a blank, a semicolon, at least one more blank, and then the reliability coefficient of the test in the form

XXXXXXXX RELIABILITY IS 0.RRR

where XXXXXXXX represents the test title (exactly as punched in the Title Card) and 0.RRR is the reliability coefficient of the test. (*Note*: The letter Ø is slashed; the number zero is not slashed.)

continued

B. SCORMACH can also process (in the same run) additional sets of score packets, each of which consists of an "anchor" packet followed by as many other packets as desired. Such additional sets of score packets should be wrapped separately and clearly marked; the anchor packet must be the first packet in each set.

Description of Output

Output from SCORMACH appears in both tables and graphs. For each test, the tables include

(a) an "echo print" of the raw scores of the examinees. This may be omitted if the user so specifies in the cover letter. (It is not included in Sample Output.)

(b) a listing of every raw score attained by the examinees in the group, each with its corresponding percentile rank and regressed score

(c) a listing of the regressed score on the *other* test that corresponds to each raw score attained by the examinees in the group

(d) a listing of the corresponding raw and regressed scores for each integer percentile rank

The graphs include

(a) the percentile ranks of the raw and regressed scores of both tests

(b) the predicted scores of each test, given the raw scores of the other test

(c) the raw scores of both tests at integer percentile values

(d) the regressed equipercentile scores—i.e., regressed scores of test one versus comparable regressed scores of test two

If CalComp plotter output is specified, a graph is produced like the final illustration.

SAMPLE INPUT

```
SAT VERB
22  3*23 25 3*26 2*27 28 29 4*30 3*31 13*32 3*33 16*34 19*35 20*36 23*37 26*38 3
3*39 27*40 43*41 34*42   49*43 47*44 28*45 33*46 41*47 27*38 29*49 28*50 34*51
21*52 23*53 13*54 10*55 22*56 14*57 11*58 5*59 5*60 9*61 11*62 4*63 4*64 2*65
3*66 2*67 3*68 3*69 2*70 71 73 ;  SAT VERB RELIABILITY IS 0.91
SAT MATH
23 4*27 6*28 2*29 4*30 3*31 14*32 2*33 12*34 10*35 7*36 8*37 18*38 18*39 15*40
27*41 42*42 33*43 36*44 34*45 51*46 40*47 41*48 36*49 31*50 28*51 34*52 20*53 24
*54 35*55 17*56 17*57 9*58  11*59 12*60 9*61 9*62 11*63 6*64 6*65 3*66 4*67 68
2*69 3*70 72 2*73 75 76 ;  SAT MATH RELIABILITY IS 0.89
```

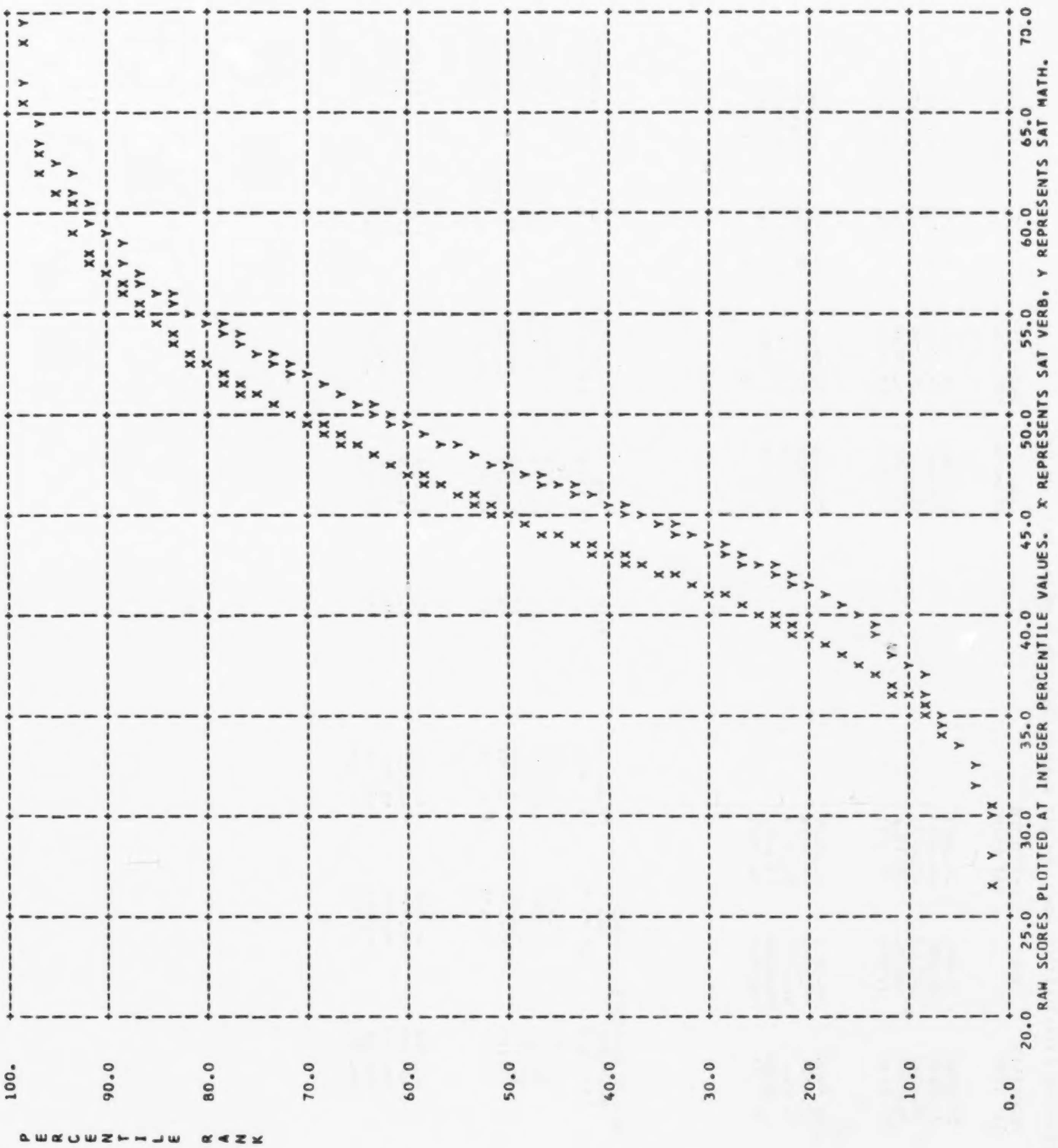
SAMPLE OUTPUT

```
             .                  .
             .                  .
             .                  .

SAT VERB

  RAW     %ILE    REGRESSED      RAW    %ILE    REGRESSED      RAW    %ILE    REGRESSED      RAW    %ILE    REGRESSED
 SCORE    RANK      SCORE       SCORE   RANK      SCORE       SCORE   RANK      SCORE       SCORE   RANK      SCORE

 22.00    0.066     23.06       36.00  10.512     36.41       49.00  67.740     48.82       61.00  94.678     60.26
 23.00    0.329     24.01       37.00  13.338     37.37       50.00  71.485     49.77       62.00  95.992     61.22
 25.00    0.591     25.92       38.00  16.557     38.32       51.00  75.558     50.72       63.00  96.978     62.17
 26.00    0.854     26.88       39.00  20.434     39.28       52.00  79.172     51.68       64.00  97.503     63.12
 27.00    1.183     27.83       40.00  24.376     40.23       53.00  82.063     52.63       65.00  97.897     64.08
 28.00    1.380     28.78       41.00  28.975     41.18       54.00  84.428     53.59       66.00  98.226     65.03
 29.00    1.511     29.74       42.00  34.034     42.14       55.00  85.940     54.54       67.00  98.555     65.99
 30.00    1.840     30.69       43.00  39.488     43.09       56.00  88.042     55.49       68.00  98.883     66.94
 31.00    2.300     31.64       44.00  45.795     44.05       57.00  90.407     56.45       69.00  99.277     67.89
 32.00    3.351     32.60       45.00  50.723     45.00       58.00  92.050     57.40       70.00  99.606     68.85
 33.00    4.402     33.55       46.00  54.731     45.95       59.00  93.101     58.36       71.00  99.803     69.80
 34.00    5.650     34.51       47.00  59.593     46.91       60.00  93.758     59.31       73.00  99.934     71.71
 35.00    7.950     35.46       48.00  64.060     47.86

SAT MATH

  RAW     %ILE    REGRESSED      RAW    %ILE    REGRESSED      RAW    %ILE    REGRESSED      RAW    %ILE    REGRESSED
 SCORE    RANK      SCORE       SCORE   RANK      SCORE       SCORE   RANK      SCORE       SCORE   RANK      SCORE

 23.00    0.066     24.36       39.00  13.141     39.45       51.00  66.886     50.77       63.00  95.335     62.09
 27.00    0.394     28.13       40.00  15.309     40.40       52.00  70.959     51.72       64.00  96.452     63.04
 28.00    1.051     29.08       41.00  18.068     41.34       53.00  74.507     52.66       65.00  97.240     63.98
 29.00    1.577     30.02       42.00  22.602     42.28       54.00  77.398     53.60       66.00  97.832     64.92
 30.00    1.971     30.96       43.00  27.530     43.23       55.00  81.275     54.55       67.00  98.292     65.87
 31.00    2.431     31.91       44.00  32.063     44.17       56.00  84.691     55.49       68.00  98.620     66.81
 32.00    3.548     32.85       45.00  36.662     45.11       57.00  86.925     56.43       69.00  98.817     67.75
 33.00    4.599     33.79       46.00  42.247     46.06       58.00  88.633     57.38       70.00  99.146     68.70
 34.00    5.519     34.74       47.00  48.226     47.00       59.00  89.947     58.32       72.00  99.409     70.58
 35.00    6.965     35.68       48.00  53.548     47.94       60.00  91.459     59.26       73.00  99.606     71.53
 36.00    8.081     36.62       49.00  58.607     48.89       61.00  92.838     60.21       75.00  99.803     73.42
 37.00    9.067     37.57       50.00  63.009     49.83       62.00  94.021     61.15       76.00  99.934     74.36
 38.00   10.775     38.51
```

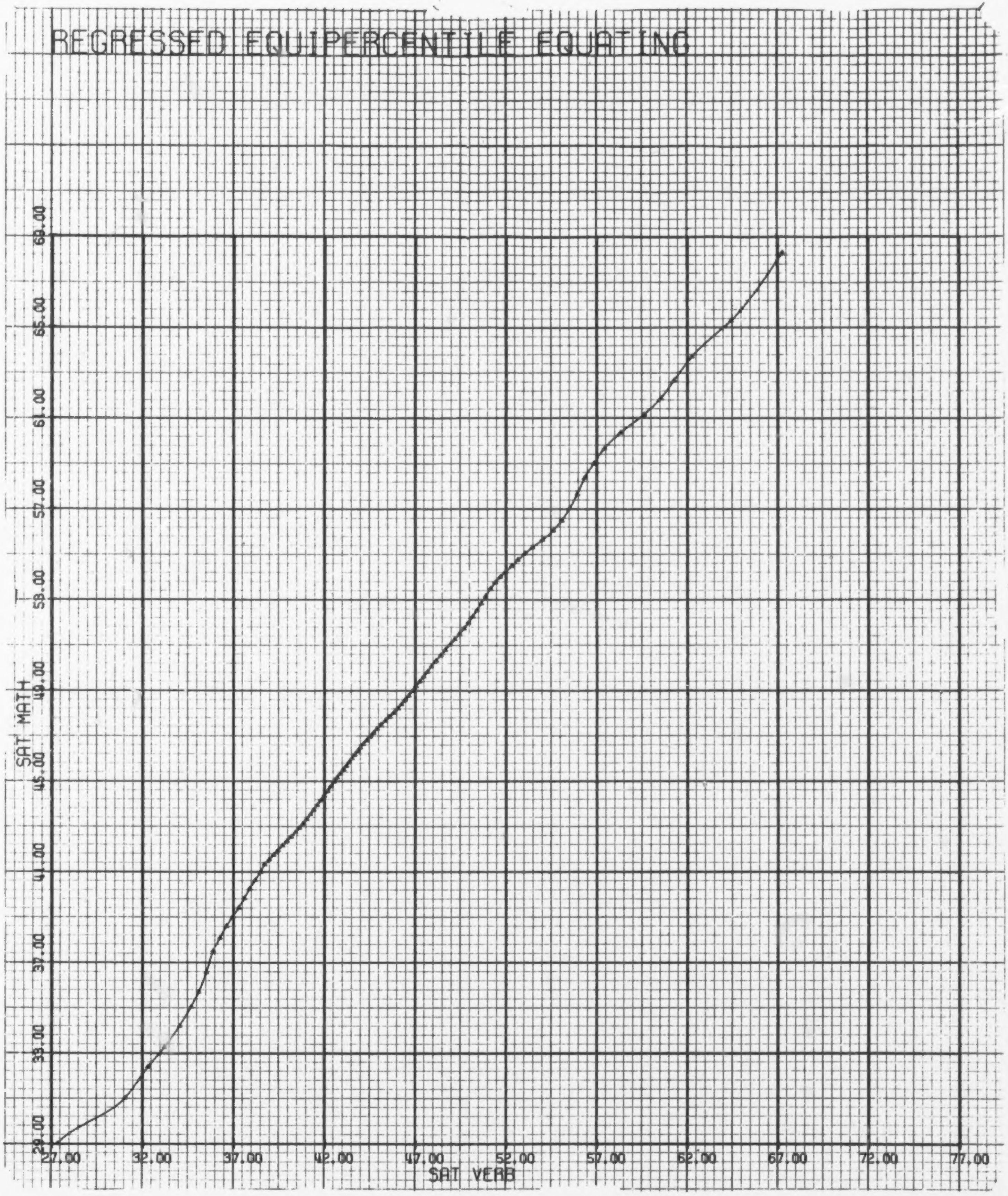
continued

```
RESULT OF INTERPOLATION AND EXTRAPOLATION FOR   50 SAT VERB VALUES AND   49 SAT MATH VALUES

     RAW       REGRESSED      %ILE      PREDICTED                          RAW       REGRESSED      %ILE      PREDICTED
   SAT VERB    SAT VERB                 SAT MATH                         SAT MATH    SAT MATH                 SAT VERB

   22.0000     23.0594       0.0657     24.3585                          23.0000     24.3585       0.0657     23.3594
   23.0000     24.0134       0.3285     27.3773                          27.0000     28.1320       0.3942     24.4903
   25.0000     25.9212       0.5913     28.4151                          28.0000     29.0754       1.0512     27.4475
   26.0000     26.8752       0.8541     28.7924                          29.0000     30.0188       1.5769     29.9277
   27.0000     27.8291       1.1827     29.3113                          30.0000     30.9622       1.9711     30.9635
      .           .             .          .                                .           .             .          .
      .           .             .          .                                .           .             .          .
      .           .             .          .                                .           .             .          .
   68.0000     66.9406      98.8830     67.9434                          72.0000     70.5849      99.4087     68.2761
   69.0000     67.8945      99.2773     69.6415                          73.0000     71.5283      99.6058     68.8484
   70.0000     68.8485      99.6058     71.5283                          75.0000     73.4151      99.8029     69.3024
   71.0000     69.8024      99.8029     73.4151                          76.0000     74.3585      99.9343     71.7103
   73.0000     71.7103      99.9343     74.3585                          -0.0        -0.0         -0.0         -0.0


VALUES COMPUTED FOR INTEGER PERCENTILE RANKS
%ILE     RAW          REGRESSED        RAW          REGRESSED          %ILE       RAW         REGRESSED       RAW         REGRESSED
       SAT VERB       SAT VERB       SAT MATH       SAT MATH                    SAT VERB      SAT VERB      SAT MATH      SAT MATH

   1     26.444        27.299         27.922         29.002             51       45.069        45.066        47.521        47.492
   2     30.349        31.023         30.063         31.022             52       45.319        45.304        47.709        47.669
   3     31.666        32.280         31.509         32.386             53       45.568        45.542        47.897        47.846
   4     32.617        33.188         32.430         33.255             54       45.818        45.780        48.089        48.028
   5     33.479        34.010         33.436         34.203             55       46.055        46.007        48.287        48.214
   .       .             .              .              .                 .         .             .             .             .
   .       .             .              .              .                 .         .             .             .             .
   .       .             .              .              .                 .         .             .             .             .
  46     44.042        44.086         46.628         46.649             96       62.008        61.225        63.595        62.656
  47     44.245        44.279         46.795         46.807             97       63.043        62.211        64.695        63.693
  48     44.447        44.473         46.962         46.964             98       65.312        64.376        66.366        65.270
  49     44.650        44.666         47.145         47.137             99       68.297        67.224        69.556        68.279
  50     44.853        44.860         47.333         47.314
```

continued


P E R C E N T I L E R A N K
100.
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
20.0
25.0
30.0
35.0
40.0
45.0
50.0
55.0
60.0
65.0
70.0
RAW SCORES PLOTTED AT INTEGER PERCENTILE VALUES. X REPRESENTS SAT VERB, Y REPRESENTS SAT MATH.

continued


REGRESSED EQUIPERCENTILE EQUATING
SAT MATH
29.00
33.00
37.00
41.00
45.00
49.00
53.00
57.00
61.00
65.00
69.00
27.00
32.00
37.00
42.00
47.00
52.00
57.00
62.00
67.00
72.00
77.00
SAT VERB

COST ESTIMATE

For each separate computer run of SCORMACH, there will be a setup charge of $50.00 plus a per-student charge of 10¢ for each test beyond the "anchor" test. Postage, handling, and consulting fees are included. And additional $20.00 charge will be made for each CalComp plot. (If CalComp option is requested, one plot will be produced for each pair of tests equated.)

Charge to user = $50.00 + 10¢/student for each test beyond
"anchor" + network overhead

CONTENTS—SCORMACH

pages

1– 2	Identification & Abstract
3– 4	User Instructions
5– 9	I/O
11	Cost—Contents

DESCRIPTIVE TITLE	Multiple Scalogram Analysis
CALLING NAME	MSA
INSTALLATION NAME	Office of Computational Sciences Educational Testing Service
AUTHOR(S) AND AFFILIATION(S)	Procedure due to: L. Guttman, J. Lingoes, University of Michigan Program due to: J. Lingoes, University of Michigan Adaptation due to: D. Kirk, Educational Testing Service Advisor on use: H. Harman, Educational Testing Service
LANGUAGE	MAP
COMPUTER	IBM 360/65
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Mr. Ernest Anastasio, Office of Data Analysis Research, Educational Testing Service, Rosedale Road, Princeton, N.J. 08540 Tel.: (609) 921-9000 ext. 2552

FUNCTIONAL ABSTRACT

This program performs a multiple scalogram analysis using the method of James Lingoes[1]. Do not confuse this with Guttman-Lingoes Multidimensional scalogram analysis which is available under the name of MSA-I (EIN Abstract 000 0070).

REFERENCES

1. Lingoes, J.C., "A Set-Theoretic Model for Analyzing Dichotomous Items," J. of Ed. Psych. Meas., *23*, pp. 501–524, (1963).

USER INSTRUCTIONS

Input Deck

Parameter Card A

Columns	*Contents*
1– 3	Number of cases to be run

Title Card

Columns	*Contents*
1	1
2–72	Any title

Parameter Card B

Columns	*Contents*
1– 3	Number of subjects ($\leq$100)
4– 6	Number of items ($\leq$122)
7	1: responses must be dichotomized 0: responses are either 1 or 0
8–10	Reproduceability criterion (normally about .80)
11–16	Chi-square criterion (normally 10.827)
17–20	ID run number (may be alphameric)
21	1: print input data 0: do not print input data
22	1: print debugging output 0: do not print debugging output
23	1: Description Cards present 0: no Description Cards
24	1: Cornell scores printed for scales with two or more items 0: Cornell scores not printed

Parameter Card C (Omit if Col. 7 of Parameter Card B is 0.)

Columns	*Contents*
1– 3	Maximum number of categories for any item

continued

Parameter Card D (Omit if number punched on Parameter Card C > 2.)

Columns	*Contents*
1– 3	Response code for item 1, response A
4– 6	Response code for item 1, response B
7– 9	Response code for item 2, response A
⋮	⋮
76–79	Response code for item 13, response B

Note: This card is used when number on Parameter Card C is 2. Puncn as many cards as required to satisfy the number of items specified in Parameter Card B (Col. 4–6).

Format Card

Standard FORTRAN format for input data. Omit the word FORMAT.

Description Deck (Omit this deck if Col. 23 of Parameter Card B is 0.)

Columns	*Contents*
1–72	Alphameric description card. Punch one for *each* item.

Data Deck

Punch one subject ID followed by all item codes for the number of items specified according to the format specified on the Format Card.

Multiple case runs will repeat the Input Deck from Title Card through Data Deck, as many times as specified on Parameter Card A.

Job Deck

The program is currently in the system and may be called as follows.

```
//JOB CARD (will be provided by ETS personnel).
//EXEC GITNGO,NAME=MSA
//GO.SYSIN DD *
    (insert Input Deck or Decks here.)
/*
```

continued

Output

The output will vary according to the options selected. The Sample Input/Output gives a hypothetical example with 8 items and 25 subjects. The output selections included input parameters, Cornell scores and answer patterns ordered by subject ID for scales 1 and 2, and subjects ordered by Cornell scores for scales 1 and 2.

SAMPLE INPUT

```
//MSATST     JOB (....
//EXEC GITNGO,NAME=MSA
//GO.SYSIN  DD    *
  1
1 THIS WILL TEST MSA WITH A HYPOTHETICAL 2-DIMENSIONAL CASE
0250080.80  3.84TEST1011
(I2,8(2X,I1))
  ITEM NO. 1
  ITEM NO. 2
  ITEM NO. 3
  ITEM NO. 4
  ITEM NO. 5
  ITEM NO. 6
  ITEM NO. 7
  ITEM NO. 8
01  1  0  0  1  0  0  1  1
02  0  0  1  1  0  1  0  0
03  1  0  0  0  1  1  0  0
04  1  0  1  1  0  1  1  0
05  1  1  0  0  1  1  0  1
06  0  0  1  1  0  1  1  0
07  1  0  0  1  0  1  0  0
08  1  0  0  0  1  0  0  1
09  1  0  1  0  1  1  0  0
10  1  0  0  1  1  1  0  1
11  1  0  0  1  1  0  0  1
12  1  0  0  1  1  1  0  0
13  0  1  1  0  1  1  0  0
14  1  0  0  1  0  1  0  1
15  1  1  1  0  1  1  0  0
16  1  0  0  1  0  1  1  0
17  1  1  0  0  1  0  0  1
18  0  0  1  1  1  1  0  0
19  1  0  0  0  1  1  0  1
20  1  0  1  1  0  1  0  0
21  1  0  0  1  0  0  0  1
22  1  0  0  1  0  1  1  1
23  1  1  0  0  1  1  0  0
24  1  0  1  1  1  1  0  0
25  0  0  1  0  1  1  0  0
/*
```

SAMPLE OUTPUT

```
THIS WILL TEST MSA WITH A HYPOTHETICAL 2-DIMENSIONAL CASE


              ORIGINAL DATA


              1 2 3 4 5 6 7 8
         1    1 0 0 1 0 0 1 1
         2    0 0 1 1 0 1 0 0
         3    1 0 0 0 1 1 0 0
         4    1 0 1 1 0 1 1 0
         5    1 1 0 0 1 1 0 1

         .        .     .
         .        .     .
         .        .     .

        20    1 0 1 1 0 1 0 0
        21    1 0 0 1 0 0 0 1
        22    1 0 0 1 0 1 1 1
        23    1 1 0 0 1 1 0 0
        24    1 0 1 1 1 1 0 0
        25    0 0 1 0 1 1 0 0
SCALE NO.      1INCLUDES THE FOLLOWING ITEMS

ITEM NO.       REFLECTED           MARGINAL SUM


    1              0                    20       ITEM NO. 1
    3              1                    15       ITEM NO. 3
    8              0                    10       ITEM NO. 8
    6              1                     5       ITEM NO. 6


CHI-SQUARE CRITERION =    3.840
PHI                  =     0.80
REPRODUCIBILITY      =    1.000
MIN. MARGINAL REPR.  =    0.700
NUMBER IN SCALE      =        4
```

continued

```
CORNELL SCORES AND ANSWER PATTERNS ORDERED
          BY SUBJECT IDENTIFICATION NUMBER
               FOR SCALE NO.  1

   ID NUMBER         CORNELL SCORE        ANSWER PATTERN
   ---------         -------------        --------------


                                           1   3   8   6
       1                   4               1   1   1   1
       2                   0               0   0   0   0
       3                   2               1   1   0   0
       4                   1               1   0   0   0
       5                   3               1   1   1   0
       6                   0               0   0   0   0
       7                   2               1   1   0   0
       8                   4               1   1   1   1
       9                   1               1   0   0   0
      10                   3               1   1   1   0
      11                   4               1   1   1   1
      12                   2               1   1   0   0
      13                   0               0   0   0   0
      14                   3               1   1   1   0
      15                   1               1   0   0   0
      16                   2               1   1   0   0
      17                   4               1   1   1   1
      18                   0               0   0   0   0
      19                   3               1   1   1   0
      20                   1               1   0   0   0
      21                   4               1   1   1   1
      22                   3               1   1   1   0
      23                   2               1   1   0   0
      24                   1               1   0   0   0
      25                   0               0   0   0   0

   COLUMN SUMS FOR THIS SCALE             20  15  10   5



  SUBJECTS ORDERED BY CORNELL SCORES
          FOR SCALE NO.  1

CORNELL SCORE          ID NUMBER
-------------          ---------


     0                     2
     0                     6
     0                    13
     0                    18
     0                    25
     1                    15
     1                     9

     .                     .
     .                     .
     .                     .

     3                    14
     3                    19
     4                    21
     4                    17
     4                    11
     4                     1
     4                     8
```

continued

```
SCALE NO.    2INCLUDES THE FOLLOWING ITEMS

ITEM NO.     REFLECTED          MARGINAL SUM

    2            1                  20        ITEM NO. 2
    4            0                  15        ITEM NO. 4
    5            1                  10        ITEM NO. 5
    7            0                   5        ITEM NO. 7

 CHI-SQUARE CRITERION =  3.840
 PHI                  =   0.80
 REPRODUCIBILITY      =  1.000
 MIN. MARGINAL REPR.  =  0.700
 NUMBER IN SCALE      =      4




 CORNELL SCORES AND ANSWER PATTERNS ORDERED

                BY SUBJECT IDENTIFICATION NUMBER

                      FOR SCALE NO.  2

     ID NUMBER           CORNELL SCORE        ANSWER PATTERN
     ---------           -------------        --------------

                                               2   4   5   7
         1                     2              1   1   0   0
         2                     4              1   1   1   1
         3                     3              1   1   1   0
         4                     2              1   1   0   0
         5                     4              1   1   1   1
         6                     3              1   1   1   0
         7                     2              1   1   0   0
         8                     1              1   0   0   0
         9                     3              1   1   1   0
        10                     2              1   1   0   0
        11                     0              0   0   0   0
        12                     2              1   1   0   0
        13                     1              1   0   0   0
        14                     1              1   0   0   0
        15                     4              1   1   1   1
        16                     0              0   0   0   0
        17                     4              1   1   1   1
        18                     4              1   1   1   1
        19                     3              1   1   1   0
        20                     1              1   0   0   0
        21                     3              1   1   1   0
        22                     0              0   0   0   0
        23                     0              0   0   0   0
        24                     1              1   0   0   0
        25                     0              0   0   0   0

     COLUMN SUMS FOR THIS SCALE               20  15  10   5
```

continued

```
         SUBJECTS ORDERED BY CORNELL SCORES
                FOR SCALE NO.  2


         CORNELL SCORE          ID NUMBER
         -------------          ---------


               0                   11
               0                   16
               0                   22
               0                   23
               0                   25
               1                   20
               1                   13
               1                   14
               1                   24
               1                    8
               2                    4
               2                   12
               2                    7
               2                    1
               2                   10
               3                   21
               3                    6
               3                    9
               3                   19
               3                    3
               4                    5
               4                   17
               4                   18
               4                    2
               4                   15

ALL POSSIBLE SCALES HAVE BEEN FORMED--MULTIPLE SCALOGRAM

          ANALYSIS IS FINISHED FOR JOB NO. TEST
```

COST ESTIMATE

The pricing algorithm for the Educational Testing Service is based on a complex formula involving the amount of core usage, card-processing, tape and dedicated device usage, lines printed and central processor unit (CPU) elapsed time. Computer costs for the job included in the Sample Input were $1.41.

Charge to user = computer costs + postage and handling + network overhead
= $1.41 + $5.00 + network overhead
= $6.41 + network overhead

CONTENTS—MSA

pages

1	Identification & Abstract
3– 5	User Instructions
7–11	I/O
13	Cost—Contents

DESCRIPTIVE TITLE	Nonmetric Multidimensional Scaling
CALLING NAME	Subroutine KRUSCAL
INSTALLATION NAME	Michigan State University Computer Laboratory
AUTHOR(S) AND AFFILIATION(S)	Alan M. Lesgold Computer Institute for Social Science Research Michigan State University
LANGUAGE	FORTRAN or COMPASS
COMPUTER	CDC 3600
PROGRAM AVAILABILITY	Decks and listings presently available
CONTACT	Dr. Anders Johanson, Programming Supervisor Applications Programming, Computer Laboratory, Computer Center, Michigan State University, East Lansing, Mich. 48823 Tel.: (517) 355-4684

FUNCTIONAL ABSTRACT

KRUSCAL is an implementation of J.B. Kruskal's recently published[1] numerical method for multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis[2]. Given a matrix of similarities or dissimilarities between *n* variables, the routine outputs a configuration of *n* points in a specified number of dimensions such that the distance between any two points is a monotone function of the dissimilarity of the two variables corresponding to those two points.

General Description of Input

KRUSCAL will accept as input any matrix or halfmatrix (below major diagonal) with or without the major diagonal, of similarities or dissimilarities, including, among others, correlation coefficients, confusion probabilities, interaction rates among groups, etc. This matrix need not be symmetric, and the program allows for missing data; interpoint distances corresponding to missing data values do not contribute to the stress.

continued

General Description of Output

The output consists of three sections for the two-dimensional case and two sections for other dimensionalities. In every case, a history of computation is printed, showing for each iteration[1] the following information.

STRESS = Normalized stress achieved[2]

SRAT = The rate of stress improvement = $\frac{\text{STRESS}_i}{\text{STRESS}_{i-1}}$

SRATAV = Weighted average of SRAT

= $\text{SRAT}_i^{1/3} \times \text{SRATAV}_{i-1}^{2/3}$

CAGRGL = Cosine of angle between gradient and previous gradient[1]

COSAV = Weighted average of CAGRGL

= $(1/3)\text{CAGRGL}_i + (2/3)\text{COSAV}_{i-1}$

ACSAV = Weighted average of the magnitude of CAGRGL, computed in the same manner as COSAV

SFGR = Scale factor of gradient, [same as Kruskal's mag(g)][1]

STEP = Step size[1]

Also, in every case, the final number of dimensions is printed. There is an option for printing the interpoint distances in each final configuration, and an option for punching the final configuration onto cards (see description of option cards).

In the two-dimensional case, the final configuration is also plotted (on the line printer, not on the plotter).

REFERENCES

1. Kruskal, J., "Nonmetric Multidimensional Scaling: A Numerical Method", Psychometrika, *29*, pp. 115–129, (June, 1964).

2. Kruskal, J., "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," Psychometrika, *29*, pp. 1–28, (March, 1964).

USER INSTRUCTIONS

Calling Sequence

The routine is called in the ordinary manner for calling FORTRAN SUBROUTINES that have no formal parameters.[1,2]

From a FORTRAN program:	From a COMPASS program:

`CALL KRUSCAL`	`BRTJ ($)KRUSCAL, ,*`
	`SLJ *+1`
	`00 DICT.`

Arguments or Parameters

All parameters are read by SUBROUTINE KRUSCAL from the data deck. See Description of Input.

Space Required (Decimal)

The subroutine itself and the subroutines that it contains take a total of 13817 words of core storage. The total amount of core storage taken up by KRUSCAL and all of the routines (including library routines) that it calls is 17285 words.

Error Returns or Error Codes

The message, KRUSCAL, IMPROPER CONTROL CARD, is printed along with the contents of any control card in the data deck that is incorrect. The program then terminates by doing a return to SCOPE through Q8QERROR.

Error Stop

A condition that is logically impossible in the program (i.e., a recognized machine error) will generate the message KRUSCAL and will terminate through Q8QERROR.

Accuracy

Accuracy is not a problem, since the original data values are stored, and the measures used in determining when a solution has been reached are functions, in part, of these original data values.

Caution to Users: Up to 60 variables may be scaled in up to 10 dimensions. Users should be aware of the "local minima" problem.[3]

continued

Flow Chart

The algorithm and logical flow of the program are described in Ref. 3. The following array names have been changed from those recorded in that reference.

In Ref. 3	*In the MSU program version*
DISSIM	DATA
DOTHER	LBLOCK

The subroutines are:

FIT	Uses latest gradient to get new configuration.
SORT	Sorts an array, dragging along one or more other arrays.
NEWSTP	Computes new value of STEP, COSAV, ACSAV, etc.
PRINDIST	Prints inter-point distances of final configuration.
KPACK	Routines for packing and unpacking IJ (see Ref. 3)
COMP	Routines to take powers and roots. Negative numbers are complemented, the root or power is taken, and they are recomplemented. Zero and first powers and roots do not use POWRF.
CCACT	Routine to convert alphanumeric parameter options into numeric parameters and to set switches based on these options.
CCIO	Routine to do numeric conversions for CCACT.

COMP, KPACK, and CCACT are COMPASS routines; the rest are in 3600 FORTRAN.

Description of Input

The input data deck should contain the following, in the order listed.

For each data set:
  Option Cards, if any.
  Configuration Deck, if desired.
  Data Description Deck.
  Data.

continued

After all the data sets (there may be more than one data set in a run), a STOP card should follow.

Option Cards

Any of the following options may be used. They may be entered in any order. Wherever a *normal case* is mentioned, it refers to the parameter values the program will assume if the particular option card is omitted. All of these options may be punched free-field, but there should not be spaces between the letters of a single word, or the digits of a single number. More than one option can be punched on a single card, but each word or number must be separated from the next word or number by a blank column. In any case, only the *first 72 columns* of the card can be used for specifying an option, and each option must be completely specified on one card, i.e., no continuations to a new card are allowed.

(a). The normal case assumes that the entire matrix of similarities (or dissimilarities) is being input. If only the lower half of the matrix (with or without the major diagonal) is being input, punch the word HALFMATRIX on an option card. Obviously, the program will then assume that the data matrix is symmetric.

(b). If the HALFMATRIX option is chosen, the normal case would be for the major diagonal to be included. If the major diagonal is not included, punch the phrase DIAGONAL ABSENT on one of the option cards.

(c). Missing data values are indicated by filling the appropriate cell of the matrix with a value that is less than the cutoff value, which is normally zero. If a different cutoff is desired, punch the word CUTOFF and a real number on one of the option cards, e.g., CUTOFF 3.0.

(d) Scaling is done using the Minkowski r-metric[4] with a normal case value of r=2.0, giving Euclidean distance. If a different value of r is desired, punch the letter R follow by a real number to specify the value of r that is desired, e.g., R 1.0 will give "city-block" distances.

(e). In the normal case, equal data values corresponding to unequal distances in the configuration *do not* contribute to the stress, and no attempt is made to equalize these distances. This is what Kruscal describes as the "primary" approach.[4] If the "secondary" approach is desired, i.e., if it is desired that unequal distances which correspond to equal data values contribute to the stress, and hence, that an attempt be made to equalize these distances, the word SECONDARY should be punched on an option card.

continued

(f). In the normal case, scaling is done for two dimensions only. Scaling may be done in any number of dimensions from one to ten. In a non-normal case, the maximum number of dimensions desired can be specified by punching the word DIMMAX followed by an integer specifying the maximum number of dimensions desired (up to ten). Similarly, the minimum number of dimensions can be changed from the normal case (two) to some other value by punching the word DIMMIN followed by an interger specifying the minimum number of dimensions desired. In the normal case, a configuration is computed for every number of dimensions from DIMMAX through DIMMIN. If this is not desired, punch the word DIMDIF followed by an integer on an option card. The scaling will then start with DIMMAX dimensions and proceed down by steps of DIMDIF dimensions until DIMMIN is passed. Thus the normal case is equivalent to the following option card: DIMMAX 2 DIMMIN 2 DIMDIF 1. The option card DIMMAX 10 DIMMIN 2 DIMDIF 3 would cause scaling in 10, 7, and 4 dimensions; note that if DIMMIN had instead been 1, scaling would have been done in 10, 7, 4, and 1 dimensions. Only one or two of the normal case values need be changed; e.g., DIMMAX 10 alone would produce scaling in 10, 9, 8, 7, 6, 5, 4, 3, and 2 dimensions.

(g). Normally, scaling for a given number of dimensions stops if STRESS is less than 0.05, if SFGR is less than or equal to zero, if SRAT is greater than 0.999, or if there have been more than 25 iterations of the scaling routine. A different lower bound can be specified for STRESS by punching the word STRMIN followed by a real number, to specify the requested boundary value, on an option card. A lower bound for SFGR can similarly be specified by punching SFGRMN followed by a real number on an option card. In the same manner an upper bound for SRAT can be specified by punching SRATST followed by a real number and for the number of iterations by punching ITERATIONS followed by an integer.

(h). If it is desired that the final configuration for each number of dimensions be punched onto cards as well as being printed, the word PUNCH should be punched on an option card.

(i). If it is desired that the interpoint distances of the final configuration be printed, the word DISTANCES should be punched on an option card.

Configuration Deck (optional)

Normally, the computer arbitrarily creates a starting configuration for the largest number of dimensions in which scaling has been requested. The results of scaling in each number of

continued

dimensions are then used as a starting configuration for the next smaller number of dimensions in which scaling has been requested. There is, however, a provision for the specification by the user of a starting configuration. A starting configuration can be used to get more iterations on a previously done scaling job, or more important, to specify a different configuration when the arbitrary configuration results in only locally minimum stress.[4] The configuration specified by the user need not have the same dimensionality as the maximum number of dimensions specified. A useful starting configuration is the principal-axis factor solution to the matrix of similarities. A configuration deck is made up in the following manner:

Card 1:	the word CONFIGURATION anywhere in Col. 1–72 on a card.	
Card 2:	in Col. 1–72, a title for the configuration—any BCD characters may be used.	
Card 3:	*Columns*	*Contents*
	1– 3	the number of points (the number of rows) of the configuration matrix
	4– 6	the number of dimensions (the number of columns) of the configuration matrix
Card 4:	*Columns*	*Contents*
	1–72	a standard FORTRAN data format for one row of the configuration matrix (This is an F-type format statement; for more information, see Ref. 2.)
Matrix deck:	the actual configuration matrix, row by row.	

Data Description Cards

Card 1:	*Columns*	*Contents*
	1–72	punch SIMILARITIES if the data are a matrix of similarities; DISSIMILARITIES if the data are a matrix of dissimilarities.
Card 2:	*Columns*	*Contents*
	1–72	the title of the data matrix; any BCD characters may be used.

continued

Card 3:	*Columns*	*Contents*
	1– 3	the number of objects to be scaled (the order of the data matrix).

Card 4:	*Columns*	*Contents*
	1–72	a FORTRAN F-type format for the longest row of the data matrix. (For more information, see Ref. 2.)

Data Deck

The data matrix should be inserted at this point. Its form will depend on whether the whole or half matrix is entered. If the whole matrix is entered, it should be entered one row at a time, with each row starting on a new card. If the half matrix is entered and the diagonal is present, then each row should include only the values up to and including the diagonal. If the diagonal is absent, the data matrix starts with row two of the similarity/dissimilarity matrix, and each row includes the values up to but not including the diagonal element.

Compute Card

After the data matrix, the data deck should contain a card with the word COMPUTE punched somewhere in Col. 1–72.

Stop Card

The last card of the job deck should have punched on it the word STOP somewhere in Col. 1–72.

REFERENCES

1. Mich. State Univ. Computer Inst. for Social Science Research, Technical Report 66–1A.

2. Mich. State Univ. Computer Inst. for Social Science Research, Technical Report 66–1B.

3. Kruskal, J., "Nonmetric Multidimensional Scaling: A Numerical Method", Psychometrika, *29*, pp. 115–129, (June, 1964).

4. Kruskal, J., "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," Psychometrika, *29*, pp. 1–28, (March, 1964).

SAMPLE INPUT

```
'JOB,015525,KRUSCAL,3.0,APHL
'EQUIP,17=(CISSR BINARY),MT(537),RO
'LOAD,17
'RUN,5.00,2000.
$ROUTINE,KRUSKAL
'MARKEF,69
'LOAD
'RUN,5.00,2000.
DIMMAX             10
DIMMIN              1
HALFMATRIX
CUTOFF           -.05
SECONDARY
ITERAT            100
DISSIMILAR
DISTANCES OF REGULAR HEXAGON
  6
(6F10.9)
0000000000
10000000000000000000
173205000010000000000000000000
2000000000173205000010000000000000000000
17320500002000000000173205000010000000000000000000
100000000017320500002000000000173205000010000000000000000000
COMPUTE
STOP
```

SAMPLE OUTPUT

```
DIMMAX           10
DIMMIN            1
HALFMATRIX       -0
CUTOFF           -0
SECONDARY        -0
ITERAT          100
DISSIMILAR       -0
DISTANCES OF REGULAR HEXAGON
  6 -0
BECAUSE THE NUMBER OF REPLICATES HAS NOT BEEN SPECIFIIED, IT IS ASSUMED TO BE 1 .
COMPUTE          -0

HISTORY OF COMPUTATION, N=    6, DIMENSION= 10

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,273  0,800  0,800  0,000  0,000  0,000  0,1134  1,5500
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        1  0,291  1,063  0,880 -0,025 -0,017  0,017  0,0892  1,3892
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        2  0,379  1,303  1,003 -0.400 -0,269  0,269  0,1078  0,6693
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        3  0,194  0,513  0,802 -0,398 -0,355  0,355  0,0967  0,3083
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        4  0,070  0,362  0,615 -0.198 -0,251  0,251  0,1043  0,1684
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        5  0,056  0,796  0,670 -0,865 -0,656  0.656  0,1112  0,0746
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        6  0,014  0,245  0,479  0,336 -0,001  0,445  0,0681  0,0349
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        7  0,006  0,441  0,466 -0,325 -0,215  0,365  0,1178  0,0205
```

continued

```
ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

SATISFACTORY STRESS WAS REACHED

THE FINAL CONFIGURATION OF   6 POINTS IN 10 DIMENSIONS HAS STRESS  0,006

FINAL CONFIGURATION
      1      2      3      4      5      6      7      8      9     10
 1  0.534 -0,471 -0,410 -0,391 -0.391  0,080  0.000  0,000  0.000  0,000
 2 -0.255  0,803 -0,351 -0,140 -0,234  0.312  0.000  0,000  0.000  0,000
 3 -0.168 -0,115  0,880 -0,200 -0,275  0.220  0.000  0,000  0.000  0,000
 4 -0.071 -0.226 -0,111  0,953 -0.144  0,071  0,000  0,000  0.000  0.000
 5 -0.077 -0,184 -0,081 -0,157  0.936  0.286  0.000  0,000  0.000  0,000
 6  0.036  0,192  0,072 -0,066  0,108 -0,969  0.000  0,000  0.000  0,000

*********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  9

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,006  0.800  0,800  0,000  0,000  0.000  0,1178  0,0356

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

SATISFACTORY STRESS WAS REACHED

THE FINAL CONFIGURATION OF   6 POINTS IN  9 DIMENSIONS HAS STRESS  0,006

FINAL CONFIGURATION
      1      2      3      4      5      6      7      8      9
 1  0.534 -0,471 -0,410 -0,391 -0,391  0,080  0,000  0.000  0.000
 2 -0.255  0,803 -0,351 -0,140 -0,234  0,312  0,000  0,000  0.000
 3 -0.168 -0,115  0,880 -0,200 -0,275  0,220  0.000  0,000  0.000
 4 -0.071 -0,226 -0,111  0,953 -0.144  0,071  0,000  0,000  0.000
 5 -0.077 -0,184 -0,081 -0,157  0.936  0,286  0.000  0,000  0.000
 6  0.036  0,192  0,072 -0,066  0,108 -0,969  0.000  0.000  0.000

*********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  8

                      .                         .
                      .                         .
                      .                         .

*********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  5

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,185  0,800  0,800  0,000  0,000  0.000  0,0329  0,3047
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        1  0.128  0,691  0,762  0,037  0,025  0,025  0,0985  0,2919
```

continued

```
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        2  0,092  0,722  0,748 -0,816 -0,530  0.547  0,0992  0,1326
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        3  0,034  0,370  0,592 -0,053 -0,216  0.221  0,0661  0,0777
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        4  0,019  0,559  0,581 -0,290 -0,265  0.267  0,1102  0,0527
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        5  0,019  0,983  0,692 -0,948 -0.716  0.716  0.1176  0,0234
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        6  0,003  0,152  0,417  0.911  0,358  0,845  0,0977  0,0139
ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

SATISFACTORY STRESS WAS REACHED

THE FINAL CONFIGURATION OF   6 POINTS IN  5 DIMENSIONS HAS STRESS  0,003

FINAL CONFIGURATION
      1      2      3      4      5
 1  0.557 -0,459 -0,410 -0,395 -0,389
 2 -0.501  0,637 -0,458 -0,178 -0.319
 3 -0.352 -0,242  0,811 -0,226 -0,341
 4 -0.042 -0,200 -0,098  0,961 -0,130
 5 -0.300 -0,346 -0,180 -0,184  0.855
 6  0.638  0,610  0,335  0,022  0,324

*********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  4

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,154  0,800  0,800 -0,000 -0,000  0.000  0.0536  0,4118
                          .                    .
                          .                    .
                          .                    .
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
       19  0,104  1,000  0,999  0,926  0.500  0,889  0,0013  0,0021
ITERATION STRESS   SRAT SRATAV CAGHGL  COSAV  ACSAV    SFGR    STEP

MINIMUM WAS ACHIEVED
```

continued

```
THE FINAL CONFIGURATION OF   6 POINTS IN  4 DIMENSIONS HAS STRESS  0,104

FINAL CONFIGURATION
     1       2       3       4
 1  0.714 -0,407 -0,387 -0,437
 2 -0.450  0,751 -0,446 -0,214
 3 -0.271 -0,173  0,917 -0,263
 4  0.008 -0,185 -0,091  0,985
 5 -0.646 -0,632 -0,350 -0,183
 6  0.645  0,646  0,357  0,111

**********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  3

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,224  0,800  0,800 -0,000 -0,000  0,000  0,0203  0,2267
                    .                          .

                    .                          .

                    .                          .
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
       37  0,010  0,946  0,936  0,212  0,112  0,186  0,0216  0,0045

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

SATISFACTORY STRESS WAS REACHED

THE FINAL CONFIGURATION OF   6 POINTS IN  3 DIMENSIONS HAS STRESS  0,010

FINAL CONFIGURATION
     1       2       3
 1  0.445  0,702 -0,555
 2 -0.468  0,875 -0,135
 3 -0.672  0,277  0,688
 4 -0.445 -0,706  0,547
 5  0.229 -0,958 -0,175
 6  0.911 -0,190 -0,370

**********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  2

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,117  0,800  0,800  0,000  0,000  0,000  0,1087  0,6382
                    .                          .

                    .                          .

                    .                          .
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        8  0,006  0,176  0,467  0,990  0,526  0,989  0,1173  0,0327

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV    SFGR    STEP
```

continued

```
SATISFACTORY STRESS WAS REACHED

THE FINAL CONFIGURATION OF   6 POINTS IN  2 DIMENSIONS HAS STRESS  0,006

FINAL CONFIGURATION
     1      2
 1  0,643  0,772
 2 -0.354  0,939
 3 -0,995  0,149
 4 -0,629 -0,766
 5  0,353 -0,934
 6  0,982 -0,159

*********

HISTORY OF COMPUTATION, N=    6, DIMENSION=  1

ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV   SFGR    STEP

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
        0  0,422  0,800  0,800 -0,000 -0,000  0,000  0,0297  0,6275
                    .                           .

                    .                           .

                    .                           .

D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
       14  0,413  1,000  0,999 -0,326 -0,033  0,554  0,0001  0,0021
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
       15  0,413  1,000  0,999 -0.997 -0.669  0,846  0,0006  0,0005
D=0A
D=0A
D=0A
D=0A
D=0A
D=0A
       16  0,413  1,000  0,999  1.000  0,432  0,948  0,0004  0,0004
ITERATION STRESS   SRAT SRATAV CAGRGL  COSAV  ACSAV   SFGR    STEP

MINIMUM WAS ACHIEVED

THE FINAL CONFIGURATION OF   6 POINTS IN  1 DIMENSIONS HAS STRESS  0,413

FINAL CONFIGURATION
     1
 1  1.364
 2  1.001
 3 -0.366
 4 -1.001
 5 -1.366
 6  0.368

*********

**********************************************
```

COST ESTIMATE

For the job listed on the Sample Output, the total processing time was 88.912 seconds. At the current rate for the Michigan State University's CDC 3600 ($245./hr.), the computer time cost $6.05 plus a charge for input and output. A 10% surcharge will be assessed if the billing is to a non-University account. In addition, there is a consulting fee of $10./hr. for work done by the Applications Programming Group.

Charge to user = computer costs + consulting + network overhead
= $6.10 (approx.) + consulting + network overhead

CONTENTS—SUBROUTINE KRUSCAL

pages	
1– 2	Identification & Abstract
3– 8	User Instructions
9–14	I/O
15	Cost—Contents